NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI) FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE

by

Chris F. Lapacik

March 1998

Thesis Advisor: E. Roberts Wood
Co-Advisor: Robert L. King

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1998	Master's Thesis

4. TITLE AND SUBTITLE
DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI) FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE

5. FUNDING NUMBERS

6. AUTHOR(S)
Lapacik, Chris F.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

A Graphical User Interface (GUI) was developed and implemented as the front end of the NPS software Joint Army/Navy Rotorcraft Analysis and Design (JANRAD). The original JANRAD computer program was developed to aid in the analysis of helicopter rotor performance, stability and control and rotor dynamics. An interactive program, JANRAD was capable of accurately and quickly solving helicopter design problems at the preliminary design level. The addition of the GUI greatly simplified the use of the program but added considerable complexity to the original MATLAB® M-File code. Because of the increased complexity, only the Performance Analysis module of the program was modified. The use of several new features of MATLAB® version 5.1, such as the GUIDE® and Structure functions, simplified the construction of the GUI environment and enhanced the tie between the user interface and performance calculation routines. Although initiated from the MATLAB® command line, the program can now be worked entirely from the "Windows" environment. The performance routines were modified extensively to connect the user input with the existing analysis routines. However, the fundamental method of analysis remains unchanged. Several cases of Sikorsky UH-60A Black Hawk input data were run and results compared with those from JANRAD version 3.1 (1995). The results correlated exactly. A Users Guide was developed and is included in Appendix A.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Helicopter, Graphical User Interface, Performance, Preliminary Design	207
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI) FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE

Chris F. Lapacik
Lieutenant Commander, United States Navy
B.S.A.E., West Virginia University, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1998

Author: _____
Chris F. Lapacik

Approved by : _____
E. Roberts Wood, Thesis Advisor

_____
Robert L. King, Co-Advisor

_____
Gerald H. Lindsey, Chairman
Department of Aeronautical and Astronautical Engineering

ABSTRACT

A Graphical User Interface (GUI) was developed and implemented as the front end of the NPS software Joint Army/Navy Rotorcraft Analysis and Design (JANRAD). The original JANRAD computer program was developed to aid in the analysis of helicopter rotor performance, stability and control and rotor dynamics. An interactive program, JANRAD was capable of accurately and quickly solving helicopter design problems at the preliminary design level. The addition of the GUI greatly simplified the use of the program but added considerable complexity to the original MATLAB® M-File code. Because of the increased complexity, only the Performance Analysis module of the program was modified. The use of several new features of MATLAB® version 5.1, such as the GUIDE® and Structure functions, simplified the construction of the GUI environment and enhanced the tie between the user interface and performance calculation routines. Although initiated from the MATLAB® command line, the program can now be worked entirely from the "Windows" environment. The performance routines were modified extensively to connect the user input with the existing analysis routines. However, the fundamental method of analysis remains unchanged. Several cases of Sikorsky UH-60A Black Hawk input data were run and results compared with those from JANRAD version 3.1 (1995). The results correlated exactly. A Users Guide was developed and is included in Appendix A.

DISCLAIMER

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Readers are also cautioned that the computer code in this thesis may not have been exercised for all cases of interest. While effort has been made, within the time available, to ensure that the program is free of computational and logical errors, additional verification should be applied. The use of this application is at the risk of the user.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

The author would like to thank Prof. E.R. Wood for his guidance, patience and the freedom to develop this software from a student's point of view. Additionally, thanks for the use of your personal computer and the latest, up to date software.

Thanks also, to LCDR Rob King for the use of his office space and countless de-bugging sessions. Some of the best MATLAB gouge available at NPS can be found in your office!

Also, thanks to LCDR Bill Hucke for his enthusiasm for this project and many hours of programming assistance. Particularly, for adding Sikorsky airfoil functions, elapsed time function and the all of the printing features. But mostly, thanks for agreeing to continue this project with so much left undone.

Many thanks to Loren Dean at The MathWorks® for his quick response to many GUIDE® and GUI questions.

To my wife, Rose, and children Joshua, Jeremy, Julaine and Joy, thank you for your love, support and the occasional reality check. This degree belongs to all of us.

To my God and Savior, thank you for this and all of the educational opportunities I have enjoyed thus far. May they all be used to your glory.

I. INTRODUCTION

A. BACKGROUND

The **J**oint **A**rmy/**N**avy **R**otorcraft **A**nalysis and **D**esign (JANRAD) computer code was originally developed in response to the 1993 American Helicopter Society (AHS) Design Competition. The emphasis on the use of locally developed analysis tools as opposed to packaged commercial software was an integral part of the Naval Postgraduate School design course philosophy. This code has been instrumental in the NPS Helicopter Design Teams winning consecutive 1st or 2nd place awards in the last 5 annual contests.

Originally developed as a preliminary design tool, the main performance function calculated 25 different helicopter performance parameters given 35 input parameters. These parameters are described in more detail in Nicholson (1993). The program was written using MATLAB® PC version 3.5 using a combination of script and function M-files. Inputs were made at the command line or loaded as a *filename*.mat file from the current working directory. The output was displayed in the main workspace and an option to save the output was also built into the program.

Since JANRAD version 1.0 was initially developed, many features have been added. A routine to analyze Stability and Control was added by Wirth (1993) and the latest, a section for Rotor Dynamics, was developed by Hiatt (1995). Minor updates have included time varying tip loss analysis, the ability to calculate performance characteristics over a range of selected input parameters and the addition of airfoil choices.

Eccles (1995) was the lastest thesis written concerning JANRAD. His thesis validated the code by a detailed comparison with Sikorsky UH-60A and H-34 measured flight test data with calculated results from JANRAD version 3.1. It was determined that JANRAD results predicted power required within 2% for altitudes below 6000 feet MSL.

These results were considered very good in light of the fact NACA 0012 airfoil data was used instead of actual Sikorsky UH-60A airfoil data.

B. JANRAD 98

During the last two AHS design competitions and numerous projects within the helicopter curriculum at NPS, students initially unfamiliar with the program described the interface as cumbersome and non-intuitive. Once familiar with the program interface however, the tool quickly demonstrated its usefulness. Since the program was difficult to run on the outset, during the design project, students would designate a specific team member to learn and run the JANRAD program throughout the design process. Because of the compressed nature of the curriculum, this was not the most efficient use of manpower and highlighted the need for an update of JANRAD. A concensus of former team members agreed that a graphical user interface (GUI) similar to Windows® would improve the overall look, feel and speed of the program. It would also allow any student to quickly learn and run models without having to refer to detailed instructions or an experienced user.

Additionally, there have been major changes to MATLAB®, the language originally used to develop JANRAD. Although graphical user interface features existed in prior versions, these features had to be coded line by line. Each user interface control such as a push button on a figure window could easily require 10 to 20 lines of MATLAB® code depending on the complexity of the interface function. MATLAB® version 5.0 included a new function that would automatically generate the required code while the programmer "Dragged and Dropped" controls or objects from a pallet to a blank MATLAB® figure window. The function called GUIDE® (Graphical User Interface Design Environment) maked the process of creating a GUI quicker and easier. The programmer now spent the major portion of his time with the task of tying the GUI to the analysis part of the program.

The MATLAB® Editor/Debugger was also instrumental in creating the GUI. This editor program automatically indented and highlighted specific MATLAB® commands with different colors. These features allowed easier and quicker identification of code and simplified the extensive modifications required in developing JANRAD 98.

With the necessary tools made available, JANRAD transitioned from a command line program, reminiscent of MS DOS, to the look and feel of some of the latest professional engineering software packages. It also gave the user flexibility and quick response to changing design specifications. The new JANRAD version 4.0 was named JANRAD 98.

C. NEW FEATURES

A basic Users Guide, included in Appendix A, describes an example run using UH-60 input data and incorporates important new features over version 3.1:

- Any or all input parameters can be changed and run without saving or altering an existing file.
- Input and output parameters are all displayed on one window.
- User can retrace steps back through program to verify selections or options.
- User can exit or quit program at any point of analysis.
- Actions such as saving or printing input/output files are executed by a single mouse click.
- Creating a new file does not require prompting for each input. Values are all typed directly into edit boxes on a single input window.
- A total of 5 airfoil choices are now available.
- User interface minimizes keyboard entries.

II. DESIGNING THE FRONT END

A. WHY CREATE A GRAPHICAL USER INTERFACE?

"Why would anyone want to create a GUI in MATLAB®?" The short answer according to Hanselman/Littlefield (1996) is probably no one. However, an engineer or engineering student might want to develop a GUI when:

- Writing code that will be use often or repetitively ; or
- When writing code that will be used mostly by other engineers or students.

Both of these reasons applied to JANRAD. It is also important to note that students have developed JANRAD primarily with other students in mind. This concept was also true for the JANRAD 98 GUI development.

B. PAPER PROTOTYPE

The first and probably most important step in developing the JANRAD 98 interface was construction of a paper prototype. Dean (1997) describes and summarizes the prototyping procedure as a low tech design strategy leading to quick iterative GUI designs. The design is done on paper using pens, tape, transparencies etc. Usability tests are run on the paper prototype by simulating the actions of the GUI using these props. By shuffling these props in response to user actions, the prototype logic is tested before any code is written.

The paper prototype used for JANRAD 98 was drawn on engineering paper using colored pencils and straight edge. The prototype showed the general layout of each proposed window and the function of the controls. At this point, no MATLAB® code had

actually been written for the GUI. Because very little was known about the capabilities or limitations of MATLAB® GUI's, the prototype started with the ideal design.

Usability tests were conducted by allowing a user to physically touch buttons on the paper prototype as if their finger was the mouse pointer. The individual windows/controls were then moved to indicate the next action of the GUI. Figure 1 from The MathWorks® shows this process with a more detailed prototype.

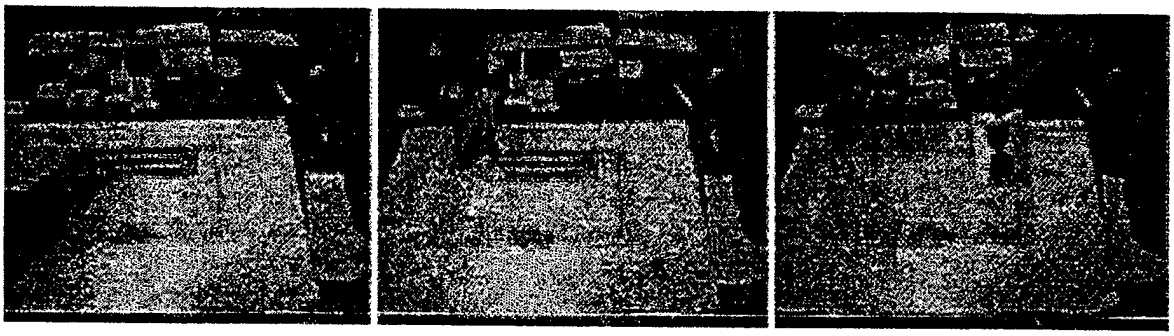

Figure 1. Paper Prototype Usability Testing at the MathWorks®.

The overall effect of paper prototyping allowed the programmer and testers to suggest and make changes quickly and without hesitation. The result was a better initial design and more confidence in the initial phase of development. It was noted that JANRAD 98 implemented 100% of the features contained in the original paper prototype plus several additional capabilities. Building GUI's with MATLAB (1997) detailed much of the GUI design principles used throughout this project.

C. USING MATLAB® GUIDE®

MATLAB®'s Graphical User Interface Design Environment (GUIDE), included in version 5.0, was the primary tool used to develop JANRAD 98. The GUIDE® function is basically a GUI used to create GUI's. It is started from the command line by typing *guide*. Figure 2 depicts the GUIDE® Control Panel. In addition to the control panel, a

blank MATLAB® figure window will also appear on start up. As shown in Figure 2, the ten different user interface controls are displayed along the bottom of the panel. These controls are generically referred to in MATLAB® code as *uicontrols*, meaning user input controls.

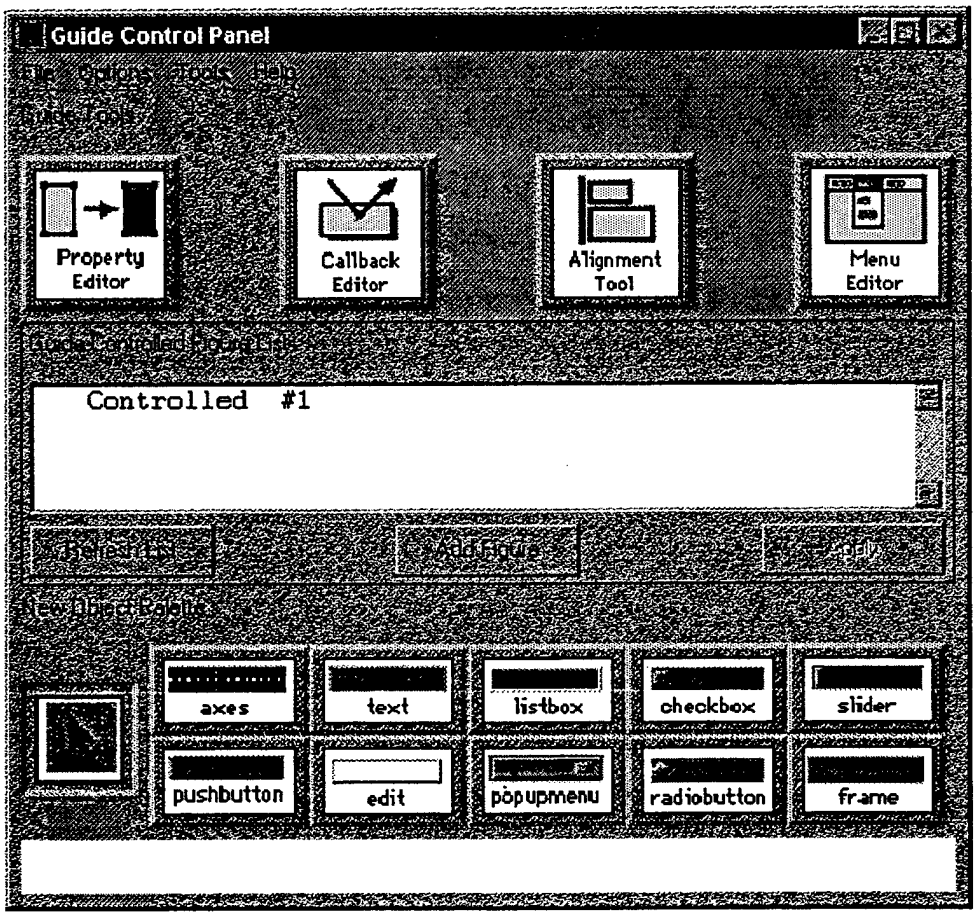

Figure 2. The GUIDE® Control Panel from MATLAB® 5.

By clicking on one of the uicontrols and minimizing the control panel, the programmer can then place the object on the blank figure window. After the desired uicontrols have been placed on the figure, they can be sized, moved and aligned using the mouse or other GUIDE® tools. Figure 3 shows a JANRAD 98 GUI window under construction using GUIDE®.

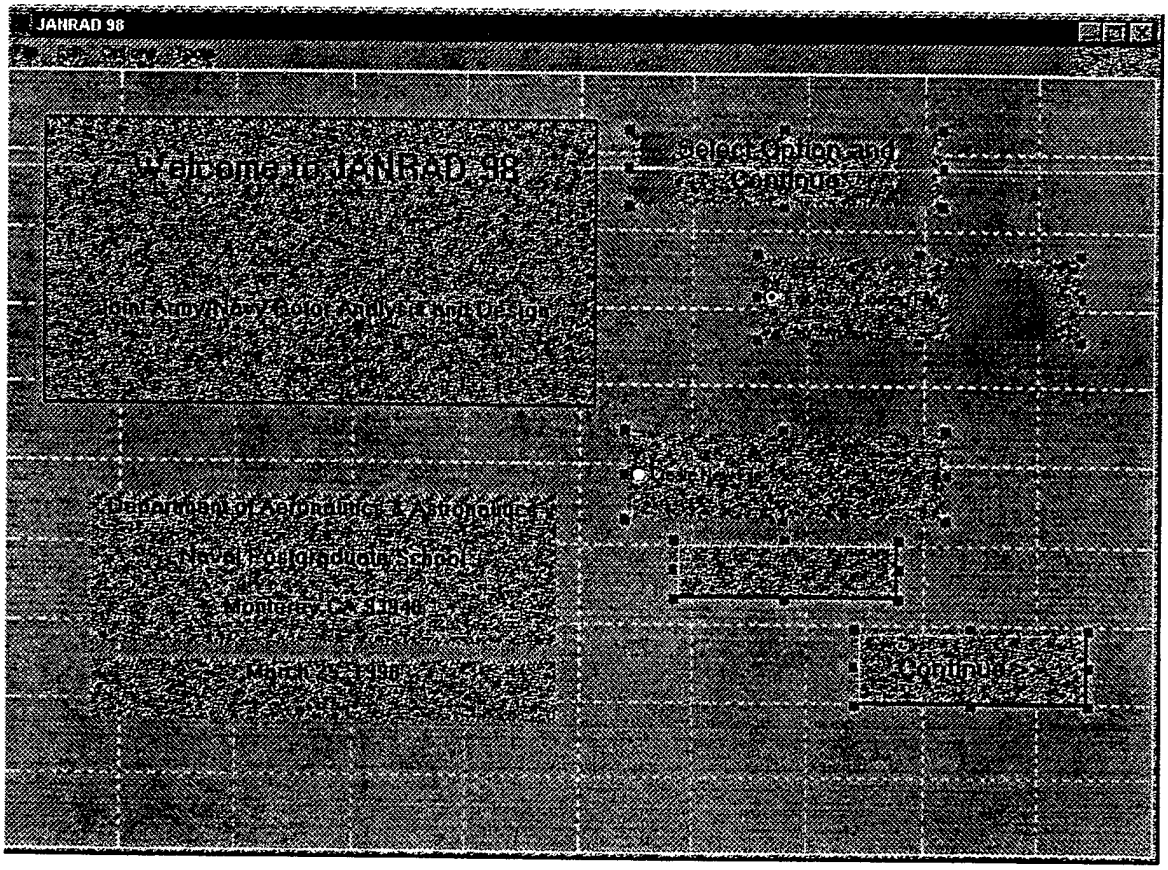

Figure 3. JANRAD 98 GUI Window Development Using GUIDE® .

GUIDE® will prompt the programmer to save the figure window when finished. At this point GUIDE® has machine generated all of the code necessary to redraw the figure when the file is executed. GUIDE® saves two files for each GUI figure. The first is a MATLAB® function with extension .m, and the second is a data file with the same filename and .mat extension. To recreate the figure, the file name is typed at the command line. The first line of the function will load the *filename.mat*. Then the remainder of the code will be executed and the figure is placed above the base workspace.

GUIDE® was also used to customize the figure by selecting specific properties for each uicontrol. These uicontrol properties determine the size, shape, color, position,

label, and even font size and weight among other things. Properties not only determine the look of the control, but also its functions. The most critical of these is the Callback property. The Callback property contains the MATLAB® commands that will be executed each time the uicontrol is activated. Any code that can be executed from the command line can be executed from the Callback. Two other related properties are the CreateFcn and DeleteFcn. These properties are similar to Callback except they will execute the commands they contain only once, when the uicontrol is either first created or deleted respectively. It is important to note here that although each uicontrol is created in a function workspace, the Callback, CreateFcn and DeleteFcn are executed in the *base* workspace. This property and its ramifications will be discussed in more detail in the next chapter.

GUIDE® generated the code for individual uicontrols in the order they were created. This often made finding a particular line within the file difficult. Particulary when doing detailed editing. In addition to the difficulty in finding specific code, while operating the GUI, the Tab key followed the order that the controls appear in the M-file. The decision was made to cut and paste the completed GUI uicontrols for all JANRAD 98 files to a more logical order. This allowed the user to use the tab key to navigate through the GUI quicker and easier. It also allowed the programmer to identify specific code wihin M-files when the GUI program was connected to the computational end.

Using GUIDE® to edit figure windows had some limitations worth noting. Specifically, when editing an existing figure window, subsequent saving deleted any customized Graphic Handles. Hanselman/Littlefield (1996) described Graphic Handles as variables attached to particular uicontrols. These handles were used extensively to change uicontrol properties during the operation of the GUI. Because of this limitation, large-scale changes to the figure window were very difficult. Small changes also necessitated a strong working knowledge of uicontrol code and Handle Graphics.

Hanselman/Littlefield (1996) and Marchand (1996) contain the most in depth discussions on the subjects of Handle Graphics.

III. CONNECTING THE BACKEND

A. FILES AND VARIABLE CONVENTIONS

To keep track of the extensive number of M-files and variables it was important to establish a convention for naming these files and variables. Many of the new files created for the interface were given longer and more descriptive file names to minimize confusion. The following lists the file name conventions:

- File names beginning with an Upper case first letter indicates a computational file. Example: *Perf.m, Trim.m, Thrcalc.m, Tmcalc.m and Dmcalc.m*
- File names in all lower case letters indicates a GUI file. Example: *janrad98.m*
- File names with lower case and _fcn.m endings indicate Switchyard Callback Functions. Example: *janrad98_fcn.m*

Because of the increased size and complexity of JANRAD 98, some conventions had to be developed to manage the additional variable names and Graphic Handles. These conventions are as follows:

- Variables in all UPPER CASE letters indicate global. Example: ASPECT
- Variables beginning with an H_ indicate a Graphics Handle. Example: H_CHECK1 indicates a global graphics handle for a check box.
- Variables beginning with an S_ indicate a structure variable. Example: S_USER_INPUT indicates a global structure.

Although not a strict convention, Graphic Handle variables were sometimes selected as the initials of the label or string that appeared next to the uicontrol in the GUI.

Particularly, this method was used on most radio buttons. For example, the handle for the Create New File radio button was H_CNF, indicating a global graphics handle.

B. THE SWITCHYARD CALLBACK

Dean (1997) describes in detail the concept of the Switchyard Callback. This is a MATLAB® programming technique that increases speed and avoids the proliferation of function files. As discussed earlier, the Callback property lists MATLAB® code that will be evaluated when the uicontrol is pushed or otherwise activated. Shorter Callback function names run faster than long lists of commands or functions. Additionally, complex Callback strings will be harder to de-bug and modify if they are nested within the GUI code itself. A Switchyard function is a single function that is called by most or all of the controls within a single figure window GUI. In other words, every control on a single window will call the same function – the Switchyard function. The Switchyard function contains a MATLAB® switch to determine which control was activated and then execute the applicable code.

C. STRUCTURES

MATLAB® version 5 incorporates new variable types, one of which is called a Structure. Simply stated, a structure is an array of data containers referred to as fields. The *struct* command is used to define the structure. The result is the ability to group a large number of variables under one name to pass to a function, assign to a workspace or declare global. This feature was use in several areas of JANRAD 98. First, the 36 separate user input variables were stored in a single structure called S_USER_INPUT. Then, after the computational routines were completed, the 25 output values defined in a structure called S_PERF_OUTPUT. Both of these structures were declared global to be available to multiple functions and Callbacks. The variables within a structure can be accessed individually or as a group. They can be displayed by typing the structure name

at the command line. In JANRAD 98, the input structure is saved in a filename.mat file, which can be loaded and edited for subsequent JANRAD evaluations. When creating a new input file, JANRAD creates the structure after the user presses the continue button in the input window.

D. MATLAB® WORK SPACES

Because functions work in their own workspace, writing complex programs with function M-files is more efficient. The single most difficult part of the interface development was ensuring all the necessary variables and graphics handles were available to the GUI Callbacks and Switchyard functions.

Initially, the plan to manage the architecture of the program was to write the entire code as functions. This included the computational files originally written as script files. As the program took shape however, this proved difficult. Because the Callback property works in the base workspace of MATLAB®, it became apparent that simply passing groups of variables from one function to the next did not work. Additionally, the Callbacks needed access to many of the graphics handles defined within the function workspace. The fix as described in Marchand (1996) required that all of the graphics handles be declared global. Many of the graphics handles, specifically the radio button handles, also needed to be assigned to the base workspace via the *assignin* command. The input and output structures were designated global to enable the user maximum flexibility to navigate freely within the GUI while being able to modify and store the input values throughout the process.

These global variables were stored in their own separate MATLAB® workspace. Once designated a global, a variable was called at the beginning of the script or function file in order to be defined. However, if a global was initially declared within a function, the value of that global became empty once that function completed its execution. In

other words, it was still a global variable, but its value was erased after the function completed its execution. This property required careful attention to when and where global variables were declared. To avoid losing the value of a global variable, some cases required the global variable to be declared within the Callback before the Switchyard function was called.

The computational files Perf.m, Trim.m, Thrcalc.m, Tmcalc.m and Dmcalc.m were called from within a Switchyard Callback Function. Because they worked within a single function workspace and shared many variables, they remained script files without significant performance degradation. The airfoil functions however, were passed a single angle of attack and a single Mach number to return a lift coefficient and drag coefficient. Therefore, they remained functions.

E. ARCHITECTURE

In general, the basic layout required the user to first start the GUI from the command line of MATLAB® by typing *janrad98*. The first GUI function executed created the first window. After making the appropriate selections, the Switchyard function changed or modified variables, closed the existing window if necessary, called the next GUI function and executed any other necessary script or function files. As the user progressed through the program, input structures were built, options saved and performance computation files were executed, all within the switchyard callback functions.

The large number of files and functions (40+) required careful management of the GUI procedures. A simple flow chart was constructed to track the files and Callback operations and is included in Figure 4. This chart describes the basic process of starting, selecting a file and running the performance routine through one complete cycle. To simplify the figure, it does not depict all possible progressions through the file structure. Appendix D through AP include all of the M-files developed for JANRAD 98.

The under lying principle in helicopter rotor design is that the main rotor forces and moments must be adjusted so the rotor provides the lift and propulsive thrust required to meet the conditions of flight. The process of adjusting the rotor's forces is referred to as "trimming" the rotor. The performance module uses a routine which employs blade element theory and a harmonic balance method for rotor trim. This method was described in detail by Nicholson (1993). The flow chart in Figure 5 graphically shows the analysis method used to complete the performance analysis.

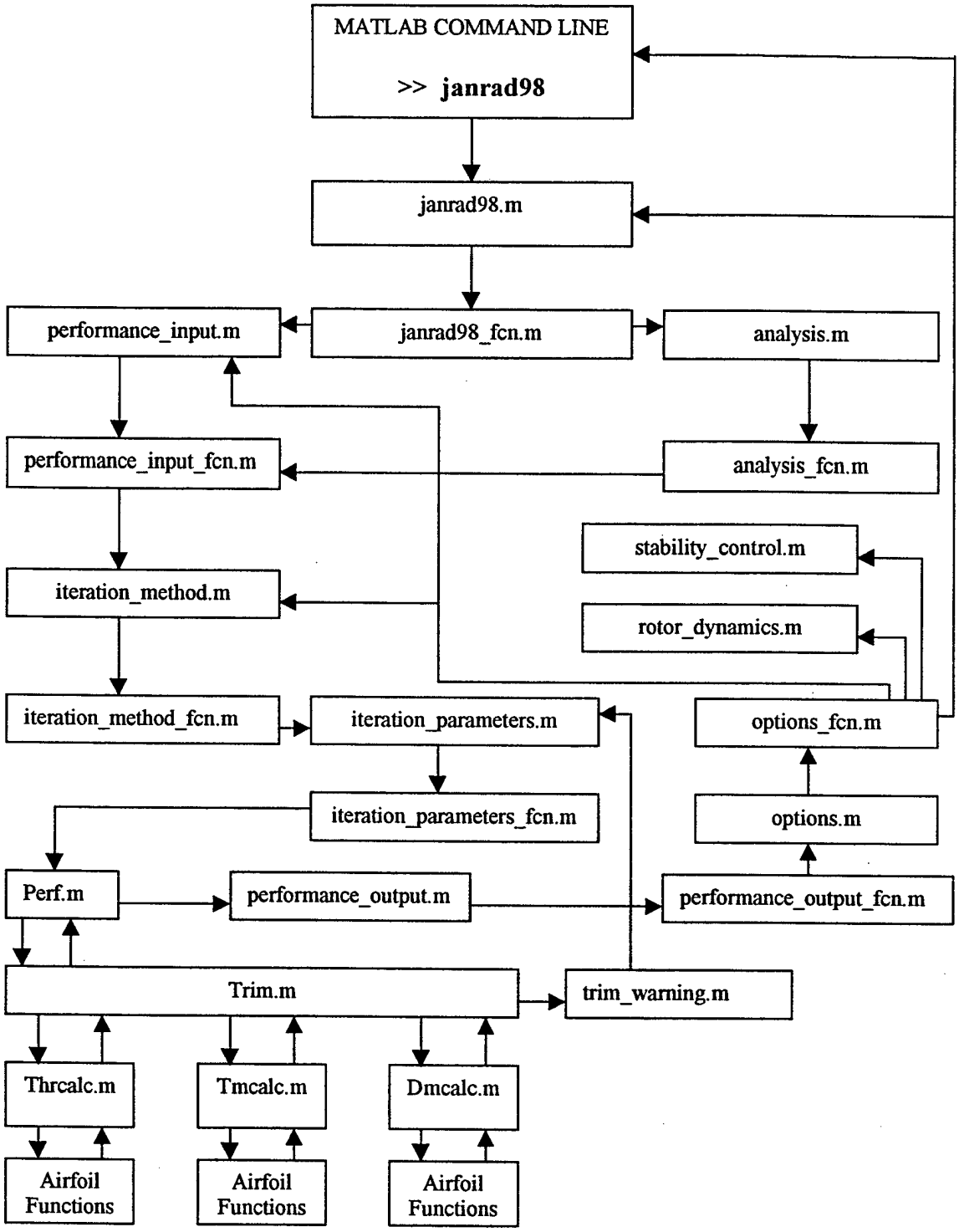

Figure 4. JANRAD 98 File Structure Flow Chart.

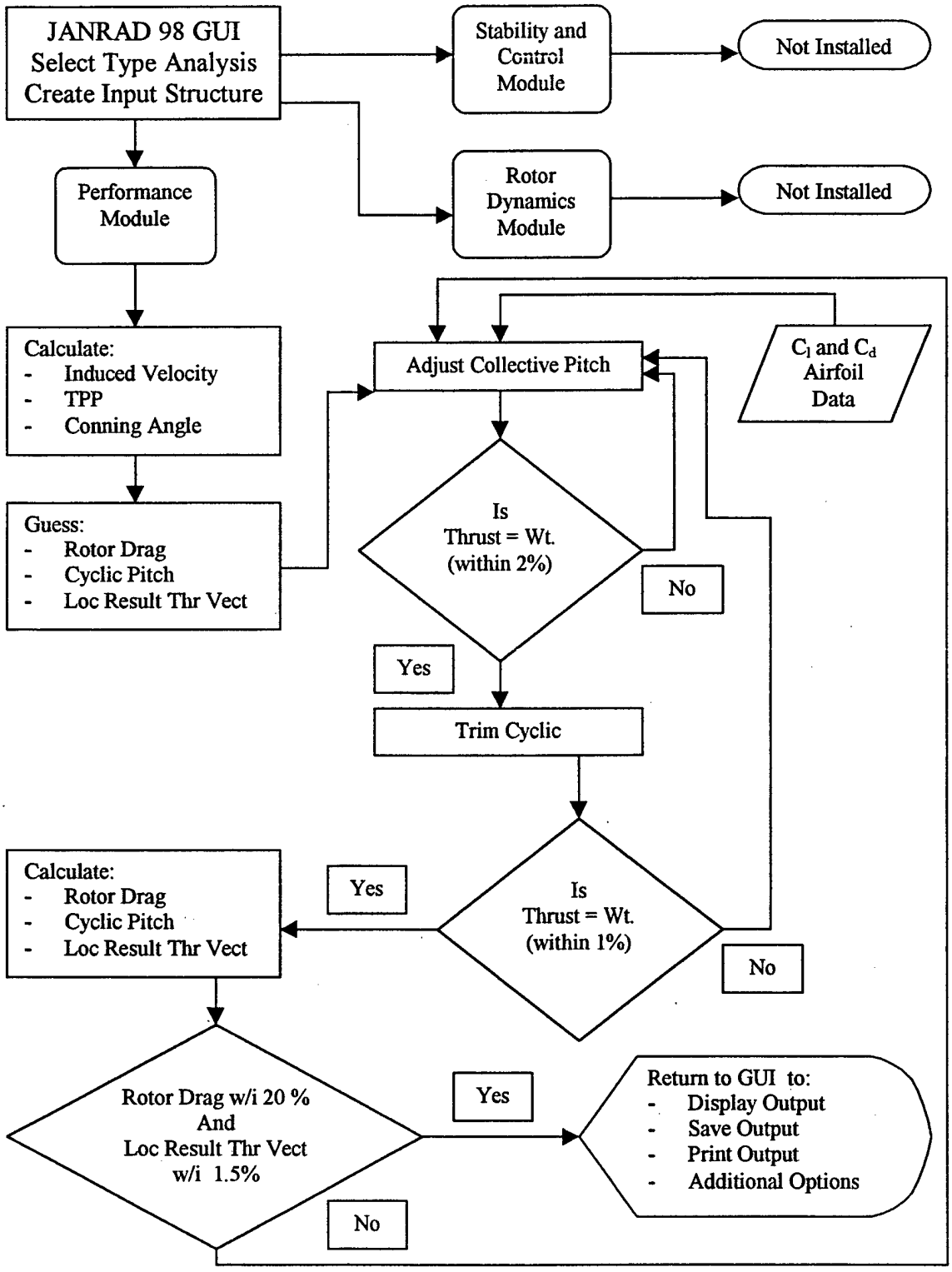

Figure 5. JANRAD Computation Flow Chart .

IV. VALIDATION

A. JANRAD 3.1/JANRAD 98 OUTPUT

Identical input parameters were run in versions 3.1 and 4.0 to validate the output of JANRAD 98. Because the format of the input was changed, a single input file to be run in either program was not possible. Therefore, two separate files were built with identical values for the input variables. As expected, the two programs calculated identical outputs. These results are contained as actual JANRAD 3.1 and JANRAD 98 printed output in Appendix B and C.

There were however, suspicions about the origin of the JANRAD 3.1 files that were modified for use in JANRAD 98. Because a master copy of JANRAD 3.1 was not available, a previously modified but undocumented version was used. It was then decided to plot JANRAD 98 power required verses airspeed and compare with flight test results as done in Eccles (1995). Figure 3 showed close correlation between JANRAD 98 and flight test data from a National Aeronautics and Space Administration UH-60A Test Flight. Again, as in Eccles (1995), VR-12 airfoil data was used in order to check the performance of the computational routines with respect to JANRAD 3.1. Figure 6 was then compared to the JANRAD 3.1 results in Figure 46 of Eccles (1995). They showed essentially identical plots. However, the exact JANRAD 3.1 inputs used to create Figure 46 of Eccles (1995) were not specifically documented. Therefore, this check served only as a general indication of version 4.0 reliability relative to version 3.1 and was not intended to substitute for a detailed code validation.

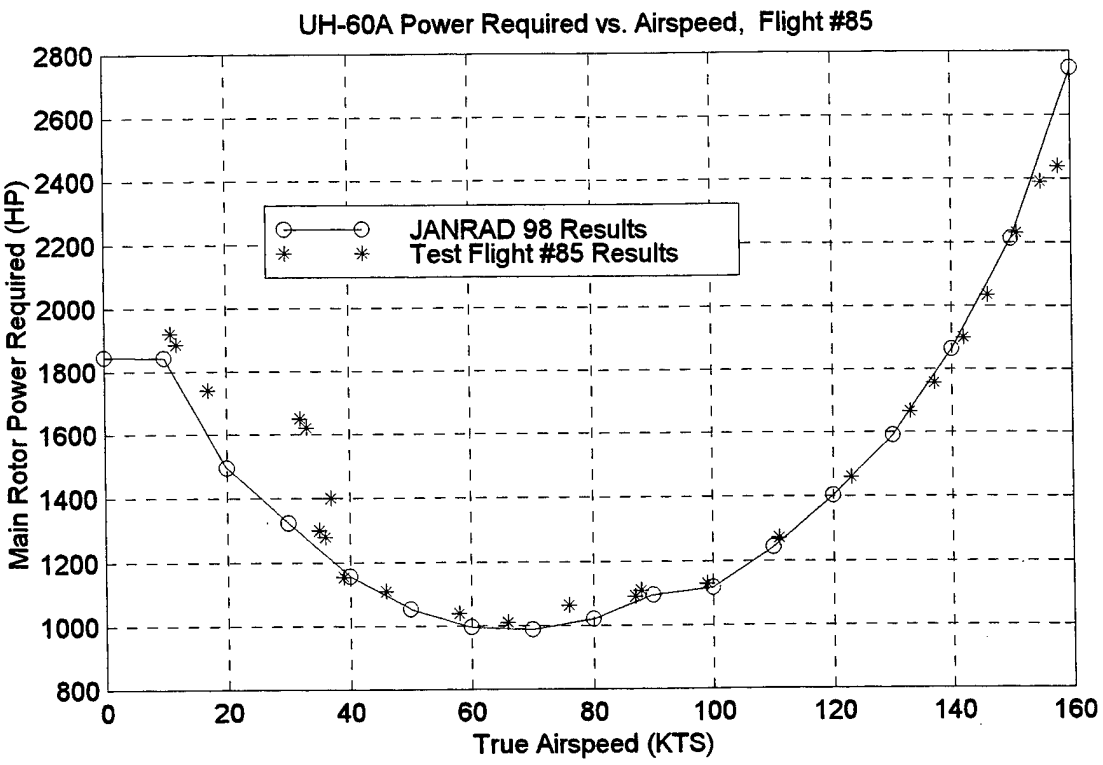

Figure 6. UH-60A Power Required vs. Airspeed, Flight #85.

B. USABILITY TESTS

Although a formal usability test was not performed, it was noted during the validation procedure that JANRAD 3.1 lacked robustness. During several JANRAD 3.1 sessions, the program locked up in what appeared to be an infinite loop. The program seemed prone to this condition particularly when editing multiple input parameters and subsequently saving these files. This was probably due to the large number of variables and evaluations taking place in the base workspace. Clearing the base workspace and restarting JANRAD usually corrected these problems. Other students have recently experienced similar problems. As discussed earlier, however, the JANRAD 3 program used may have undocumented modifications that caused these complications.

JANRAD 98 on the other hand, demonstrated superb robustness. During a single continuous session, the program ran several new and edited files as well as several input iteration runs without any apparent problems. Saving and printing operations performed well. Additionally, running several consecutive inputs proved much quicker and easier. The program could be toggled from output back to the input with a click of the mouse button. Subsequent computations were started with two additional mouse clicks.

V. RECOMMENDATIONS

Having demonstrated the usefulness of the graphical user interface, there are unlimited possibilities to improving the look, feel and features of JANRAD 98. Recommendations are subdevided into three areas: (1) GUI recommendations; (2) analysis recommendations; and (3) general recommendations.

A. GUI RECOMMENDATIONS

First, the GUI should continue to be modified according to student/user preferences. If JANRAD 98 is being used differently than the author's original intent, these changes should be reported to the Helicopter Design Team instructor, who inturn will document the necessary modifications. Enabling and disabling uicontrols within the program will need careful attention. Although every attempt has been made to prevent the user from making illogical selections within the GUI, it is impossible to eliminate all possible errors. If any of these errors are identified they should be corrected by disabling the appropriate uicontrols or the development of warning windows to notify the user of the limitations of the program.

Presently, only the airspeed iteration method will plot a series of performance calculations with respect to the airspeed variations. The addition of plotting routines for the other iteration methods would allow for a more compete analysis, specifically when trade-off studies are desired. An additional window allowing the user to select 4 or 5 specific pre-determined plots or subplots via check boxes would improve the program greatly. An 'Other' check box could refer the user to an empty script M-file to modify for custom plots of special interest without having to modify the performance files.

Because most student engineers would not be familiar with shapes of particular c_l and c_d curves for various airfoils, saved plots of these curves, available for viewing, would enhance the users ability to pick an appropriate airfoil. These curves could be plotted and their resulting figures saved to avoid having to execute a script file each time an airfoil selection is examined. The selection could be made though a separate pull down menu titled *Airfoil Plots* appearing above the input window.

B. ANALYSIS RECOMMENDATIONS

The analysis routines developed for JANRAD 3.1 and used in JANRAD 98 were the same ones validated in Eccles (1995). Although the results were generally good, there were some unexplained variations at altitudes above 6000 feet. With the role of helicopters expanding to include high density altitude operations, this limitation is significant. A similar analysis using the Sikorsky SC1094r8 and SC1095r8 airfoil data currently being added to JANRAD 98 should be completed to gather more insight to this documented discrepancy.

Additional airfoil data from a variety of manufactures and type/class helicopters would allow users to better match the mission and type helicopter with appropriate airfoils. As a minimum, data should be included for a large transport, military attack, and a light civil helicopter.

The performance routines currently assume the center of gravity of the aircraft to be located directly below the rotor mast. The capability for the user to offset the center of gravity from the rotor hub would greatly enhance the capabilities of the program. This capability should be included as a single user input as well as an iteration parameter to study the effects of CG travel on performance and stability and control.

A 3-D plot of airloads or induced velocity at selected azimuth intervals or at least at $\psi = 0°, 90°, 120°$ and $180°$ should be produced as part of the output. These plots would improve the validation of each successive run. Should a trim routine not properly converge or have other errors, it would be immediately apparent by the shape of this plot.

C. GENERAL RECOMMENDATIONS

First, master copies of JANRAD 98 should be maintained and clearly marked. Any subsequent modifications should be saved on separate disk and clearly marked as an updated or modified version. JANRAD 98 as described by this thesis is version 4.0. Any modification hereafter should be clearly labeled version 4.1 or 5.0. A general explanation should be included as comments preceding the code in the janrad98.m file. Additionally, the author's name should be added to the *About JANRAD* window found in the about_janrad.m file. If the modifications or additions were not part of a thesis, a detailed description of the theory and code should be included as an attachment to this thesis. Also a binder should be maintained with all attachments and theses related to JANRAD.

To increase exposure to JANRAD 98 and to maximize user feedback, the program should be available for download on the MATLAB® ftp site. Include Appendix A, JANRAD 98 Users Guide, as a Readme.txt file for those unfamiliar with its use. An email to select industry and university rotorcraft engineers announcing JANRAD 98's posting on the site should also be considered.

APPENDIX A. JANRAD 98 USER'S GUIDE

The JANRAD 98 Users Guide is written as a brief introduction to the Joint Army/Navy Rotorcraft Analysis and Design computer program. It is intended to explain the basic features and operation of the program and assumes a basic knowledge of helicopter mechanics and the use of the MATLAB® programming language by The MathWorks® Inc.

A. SYSTEM REQUIREMENTS

JANRAD 98 requires MATLAB® version 5.0 or MATLAB Student Edition version 5.0 or higher. It will not run on any previous versions. JANRAD 98 version 4.0 will fit on a single 1.44 MB floppy disk and will need that much memory available for installation. JANRAD 98 requires only the hardware to support MATLAB® 5.

B. INSTALLATION

The recommended installation of JANRAD 98 is accomplished by first creating a subdirectory of MATLAB called Janrad98. The entire contents of the JANRAD 98 floppy disk should be copied into this directory. Include all M-files and .mat files. JANRAD 98 will not run without all of the .mat files.

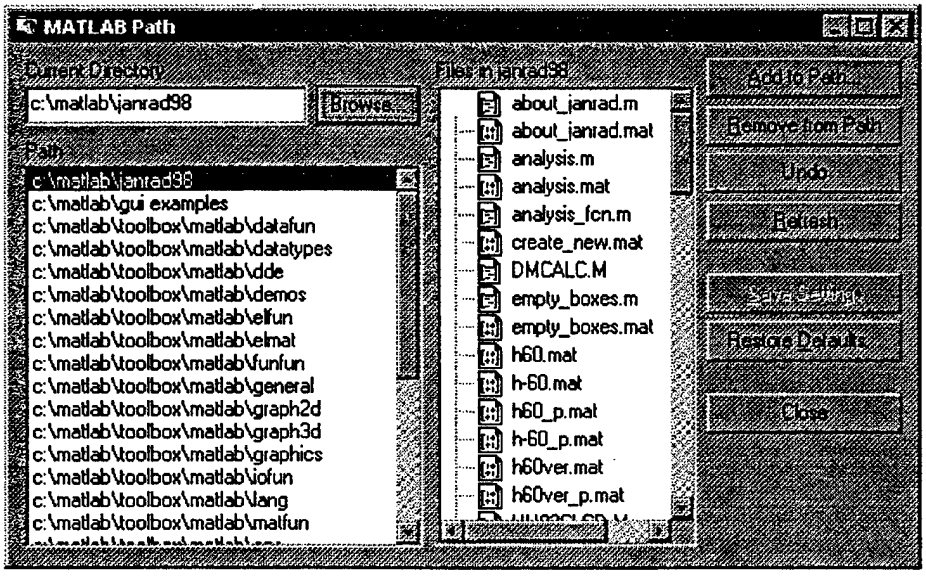

Figure A.1. MATLAB 5 Path Window.

It is recommended to add this new subdirectory to the MATLAB 5 search path. This procedure will eliminate the need to change the working directory from the command line each time JANRAD 98 is run and allows you to work from a floppy disk if desired. Adding the subdirectory to the search path is accomplished by selecting *File*, *Set Path*... from the File menu. Change the current directory to the new Janrad98 subdirectory by using the *Browse* button. Then press the *Add to Path* push button. You will then be given the option to save the new path or just use the new path for the active session. It is recommended to save this path. Figure A.1 shows the MATLAB® Path window.

C. STARTING JANRAD 98

Typing janrad98 (lowercase, one word) at the command line prompt of a current MATLAB session starts JANRAD 98. This action will launch the JANRAD 98 welcome window shown in Figure A.2.

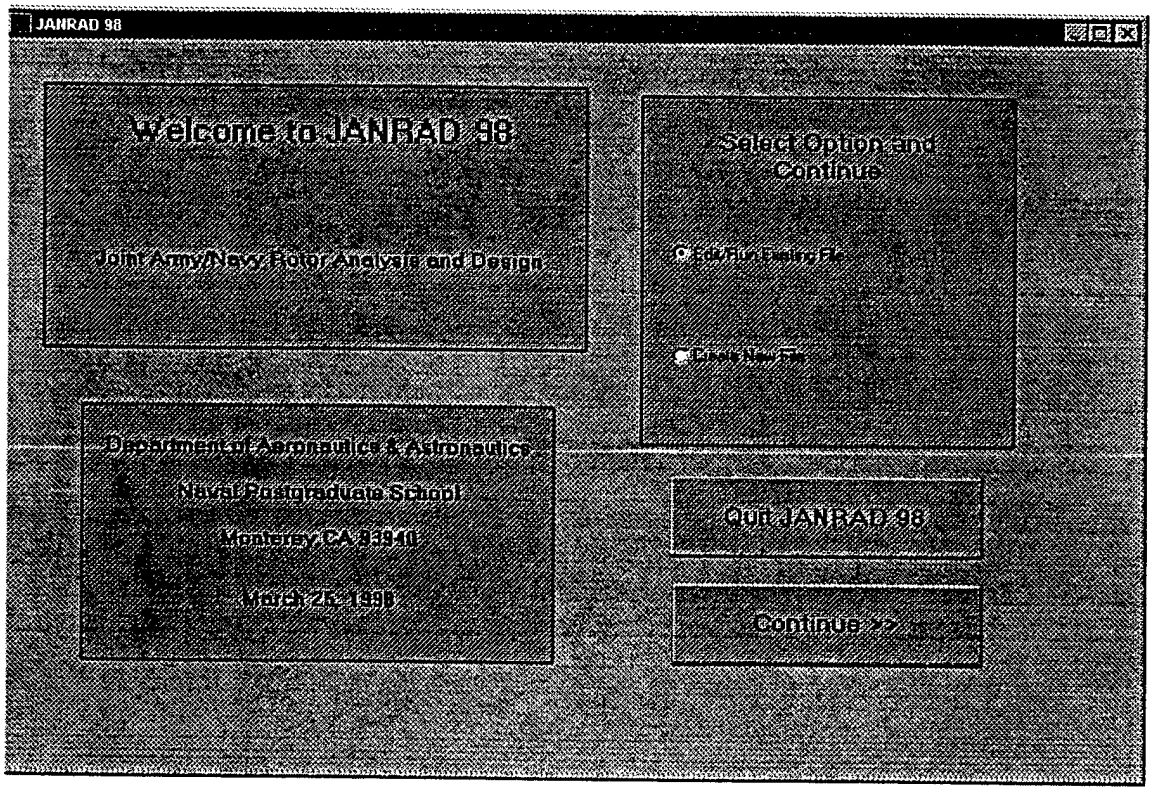

Figure A.2. JANRAD 98 Start Up Window.

D. USING JANRAD 98

As an example, the use of this program will be demonstrated by selecting a previously saved input data file and changing the weight, airspeed and pressure altitude. The input and output files will be saved and printed. After the performance analysis is complete, we will then iterate on airspeed from 80 to 100 knots in increments of 5 knots.

First, from Figure A.2, select the Run/Edit Existing File radio button. It is usually easier to edit an existing file because Create New File will not give you the chance to change the working directory if desired. Once the selection has been made, press the *Continue* >> button.

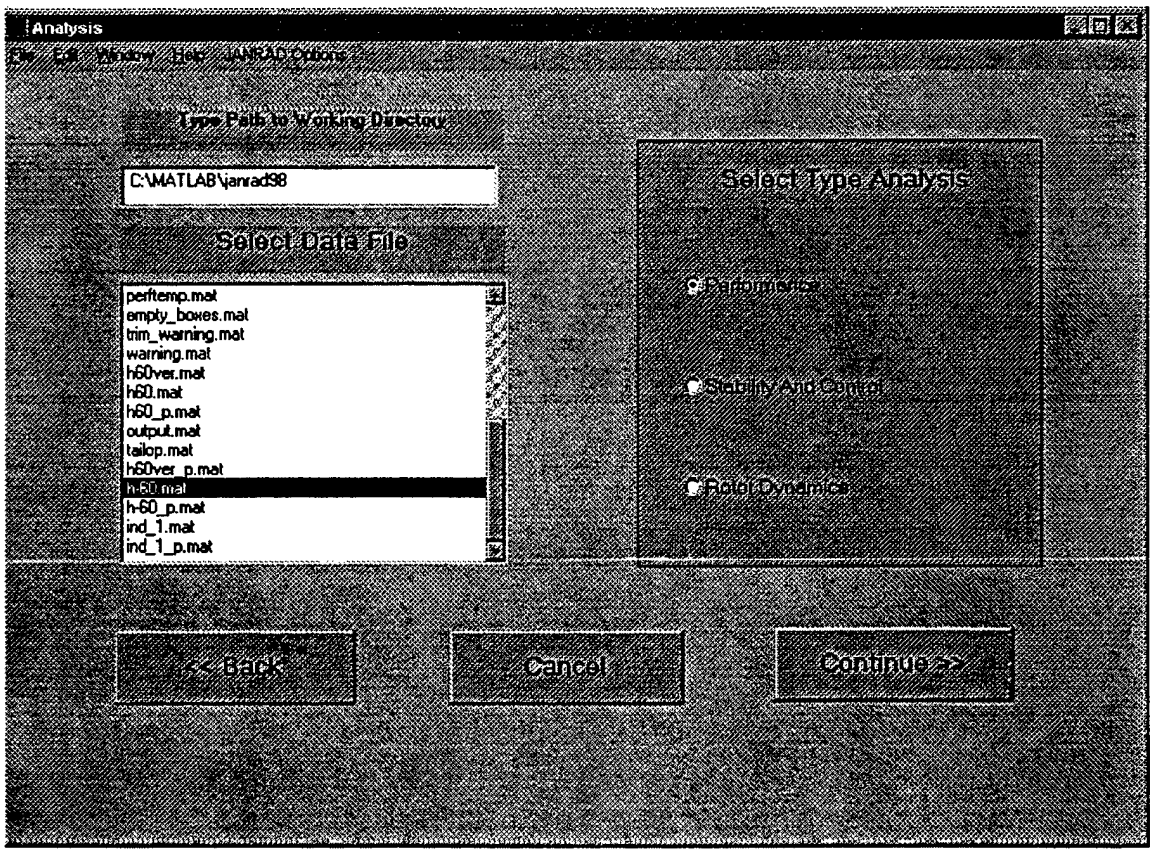

Figure A.3. Selecting a File to Edit.

The next figure window to appear is shown in Figure A.3. This window allows you to change the working directory and select an input data file. To change the directory, type or edit the desired path in the edit box. All of the .mat files listed in the

working directory are displayed in the list box. Input data files are saved as *filename.mat*. A note of caution here, each GUI window also has an associated *guifilename.mat* file. The user should name input/output data files using helicopter aircraft designations such as UH-60A, h-99 or h-design1 to differentiate from JANRAD 98 GUI files. Next, select an input file to edit by clicking on the file name. For this example, we will edit the UH-60A.mat file. Then press *Continue* >>.

The Performance Input window will be displayed as in Figure A.4. The input data will be displayed within the appropriate edit boxes. Any or all of the parameters can be changed at this point without altering the original data file. You will have the opportunity to save the new data if you chose after the analysis has been completed. The airspeed, weight and pressure altitude can be edited by highlighting and typing 100, 17,000 and 2,000 in the respective edit boxes. Pressing the enter key is not necessary to enter the new value. Using the Tab key or clicking on another edit box or control will enter any changes. The Print Screen button will print a draft copy of the GUI window with the displayed values. It however, will not record the file name for which the values are stored. Now press the *Continue* >> button.

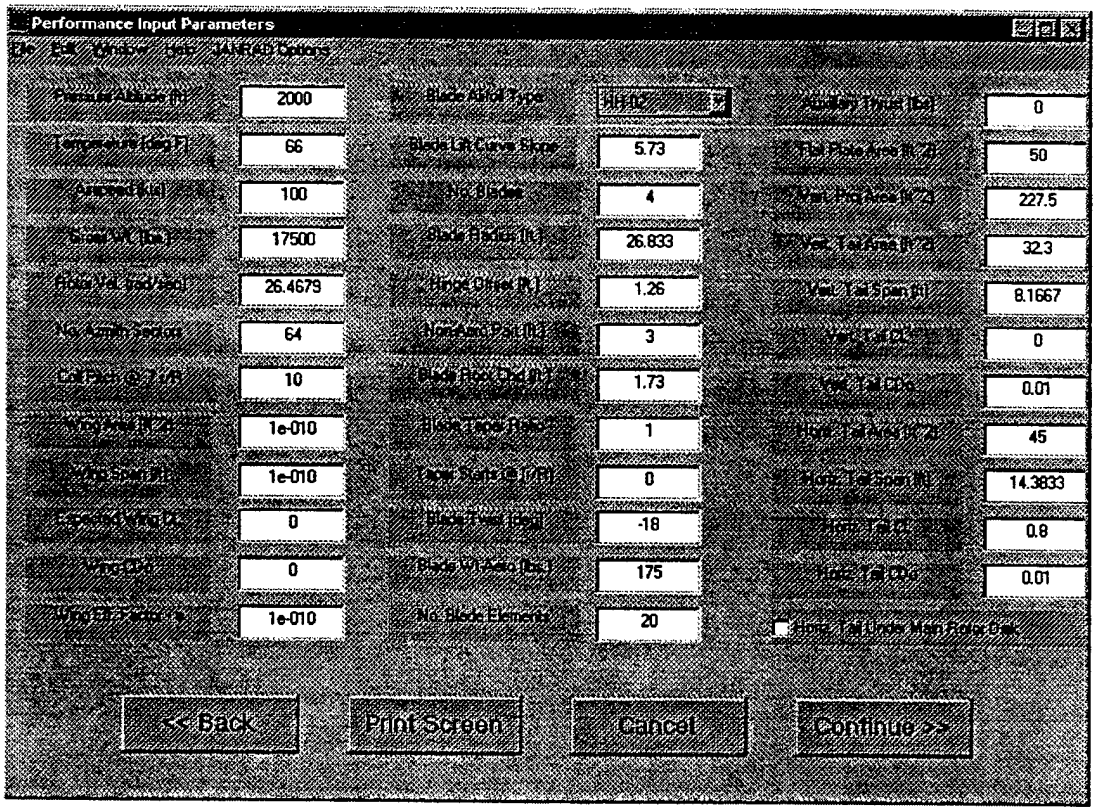

Figure A.4. Input Parameters Window.

From the Input Parameters window, JANRAD 98 will call the Iteration Method window as shown in Figure A.5. JANRAD 98 will call its computational routines from this window. By choosing *No Iteration* and *Analyze*, JANRAD 98 will run the parameters selected from the previous window. The *Analyze* pushbutton initiates the computational routines. All controls on the GUI will be disabled except the Interrupt pushbutton. The Analysis Status Box will display the performance routine status and clock as JANRAD trims the rotor and adjusts the collective and cyclic mathematically. The Interrupt button will halt the routine and enable the Resume control and JANRAD Options menu on the GUI. This will allow the user to change parameters, quit or return to beginning. The Resume button will continue with the performance routine where it originally interrupted. It is worth noting that the Interrupt button will not always respond immediately. However, once MATLAB finishes its current line evaluation, the calculation will pause.

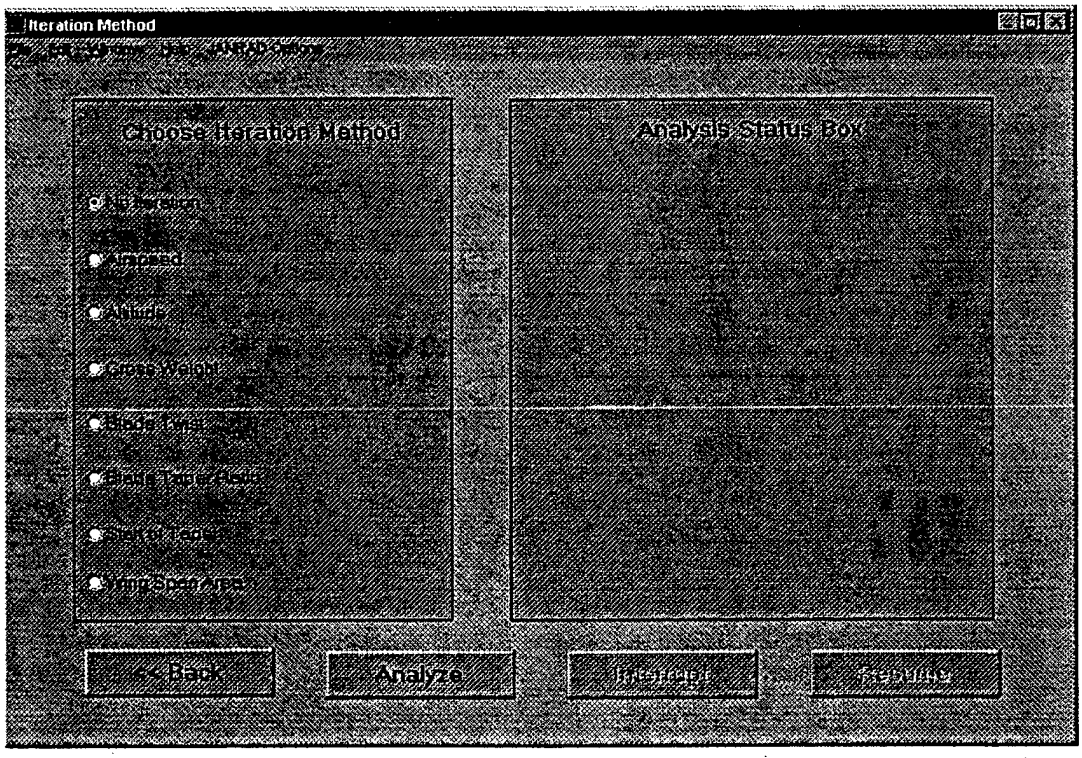

Figure A.5. Iteration Method / Analysis Window.

The Status Box will inform the user when calculations are complete. The Performance Output window will be displayed automatically. The Performance Output window shown in Figure A.6 displays the performance results. These results can be saved and the screen printed from this window. However, it is recommended to print the

saved input and output files through the next window. By pressing the *Options* >> push button, the saved input and output files can be printed simultaneously and in a more usable format. The input/output files are saved after activating the checkboxes, typing a file name and pressing the *Options* >> push button.

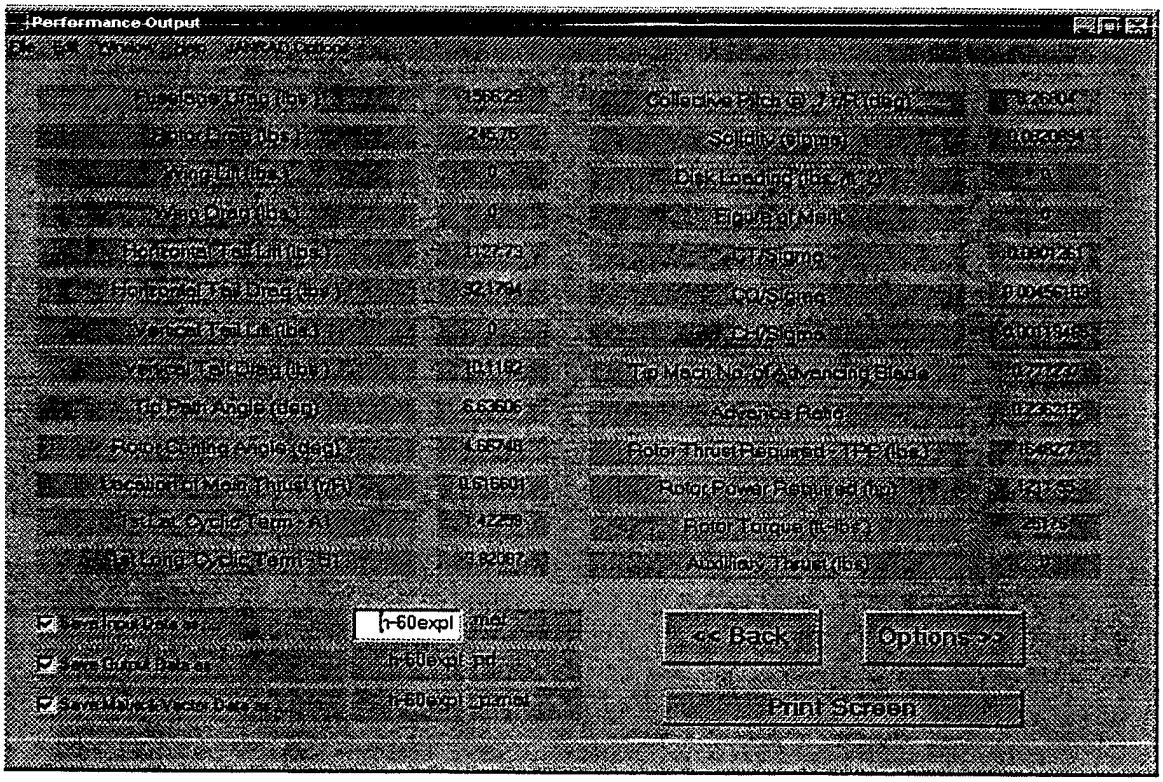

Figure A.6. Performance Output Window.

The Options window in Figure A.7 will provide the capability to print the latest files and select additional analysis routines. At this time however, the Stability and Control and Rotor Dynamics routines have not been completed. If selected, you will be reminded of this limitation.

From here, we will go back to calculate performance parameters by varying airspeed. This is done by pressing the *Change Iteration Method* radio button and the *Continue* >> push button. This will go back to Figure A.5. To vary airspeed, press the *Airspeed* radio button and then *Analyze*.

Figure A.8 shows the Iteration Parameters window. By typing 80, 100 and 5 in the appropriate edit boxes and selecting *Analyze*, the performance routine will calculate and display plots of various performance results with respect to airspeed. Note: the *Aspect Ratio* edit box and *HIGE* check box are only enabled when *Altitude Iteration* or *Wing Span Area* is selected. Figure A.9 shows an example an airspeed iteration subplot. These plots are primarily used to examine trade off studies during the design process.

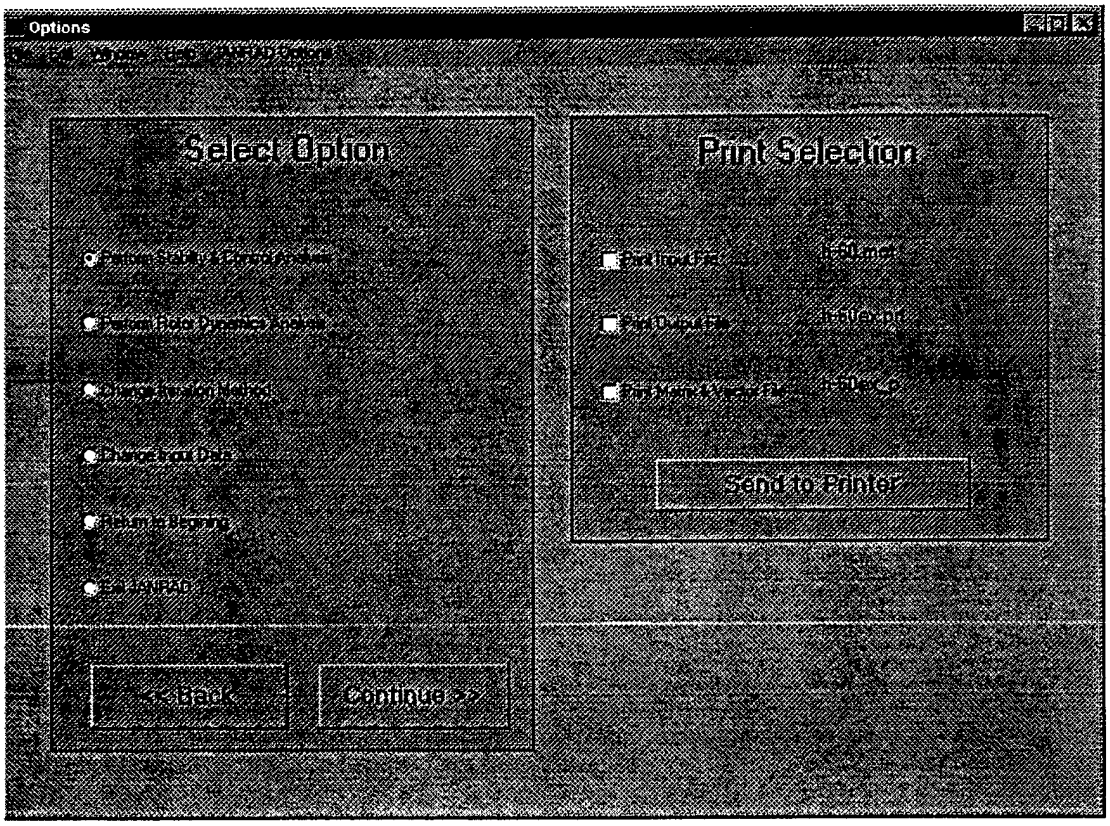

Figure A.7. Options Window.

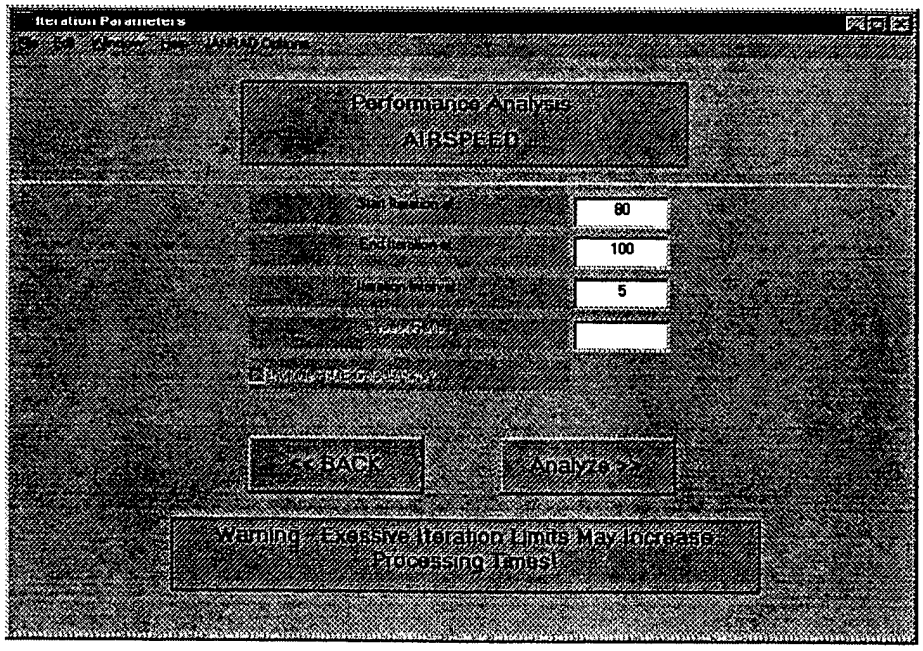

Figure A.8. Iterations Parameter Window.

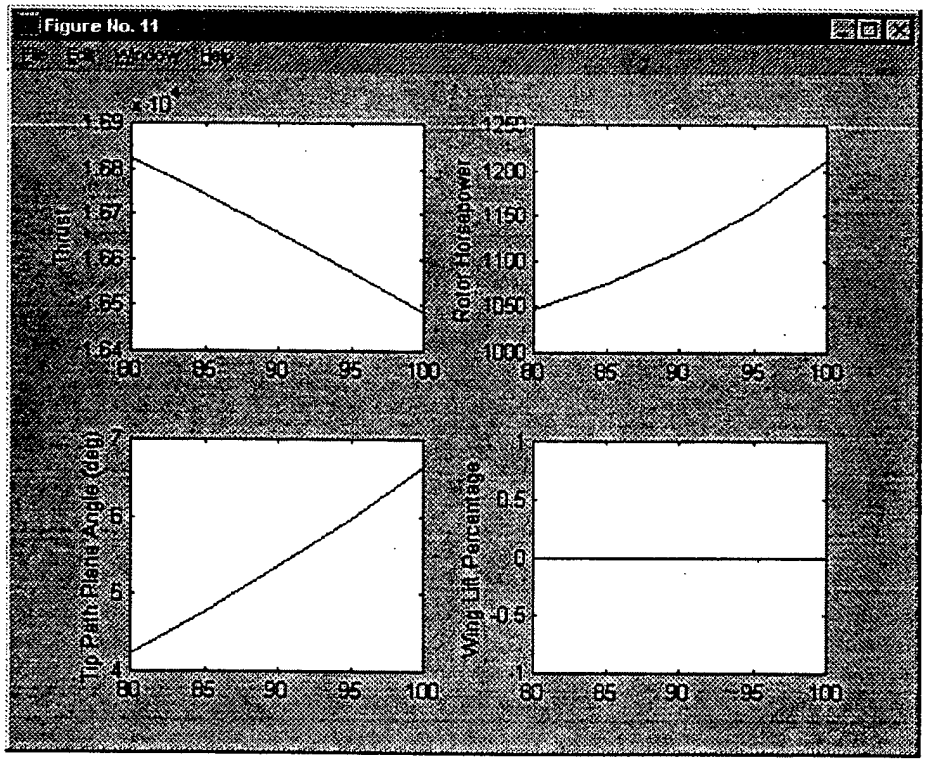

Figure A.9. Example of Airspeed Iteration Output Plot.

The M-file janrad98_plots.m contains the details for the airspeed iteration plots. You can add, change or otherwise personalize the plots within this file as necessary. It is, however, recommended that any plots not desired be eliminated by commenting the appropriate lines and adding any additional code at the end of the file to preserve the original file structure. Always document changes made to the code for future users.

E. HINTS FOR JANRAD 98 OPERATIONS

The following paragraphs list some recommendations for the most efficient use of JANRAD 98. They are a guide based on observation, experience and knowledge of the code. Any other recommendations should be addressed to the Helicopter Design Instructor at the Naval Postgraduate School for implementation into the next version of JANRAD 98.

JANRAD 98 was designed for robust operations. However, because this is the first version to utilize a Graphical User Interface, not all cases of user inputs have been exercised. If the program appears not to be working properly, quit JANRAD using the options menu available on all but the first window. This action will quit JANRAD, close all MATLAB figure windows and clear the base workspace. Restart JANRAD by typing janrad98 at the command line.

Using azimuth sectors greater than 24 and blade elements greater than 20 will dramatically increase computation time. The accuracy of the results does not improve a significant amount for these larger values.

For more accurate results, use 0012, sc1094r8 or sc1095r8 airfoils when possible. These airfoils include Mach number inputs for c_l and c_d calculations. The HH-02 and VR-12 data do not depend on Mach number.

Using the print screen buttons on the performance input or performance output windows will take up to several minutes on older processors. Use the Send to Printer button on the JANRAD 98 Options page for faster and more compact printer output.

APPENDIX B. JANRAD 3.1 INPUT/OUTPUT

The following is the actual h60.prf" file from JANRAD 3.1 used to compare with JANRAD 98 output.

```
                *** RESULTS ***
                    h60

            Forward velocity =    115 kts
                 Temperature =     66 degs F
           Pressure altitude =   1000 ft
                Gross weight =  16500 lbs
            Number of blades =      4
                Rotor radius =  26.83 ft
            Blade mean chord =   1.73 ft
                 Blade twist = -18.00 degs
     Blade lift curve slope =   5.73
                Blade weight = 175.00 lbs
        Rotational velocity =  26.47 rads/sec
          Blade grip length =   3.00 ft
                Hinge offset =   1.26 ft
  Equivalent flat plate area =  45.00 ft^2
     Vertical projected area = 227.50 ft^2
                  Wing area =   0.00 ft^2
                  Wing span =   0.00 ft
                    Wing CL =   0.00
                   Wing CDo = 0.0000
     Wing efficiency factor =   0.00
       Horizontal tail area =  45.00 ft^2
       Horizontal tail span =  14.38 ft
         Horizontal tail CL =   0.80
        Horizontal tail CDo = 0.0100
         Vertical tail area =  32.30 ft^2
         Vertical tail span =   8.17 ft
           Vertical tail CL =   0.00
          Vertical tail CDo = 0.0100
              Fuselage drag =   1927 lbs
                 Rotor drag =    336 lbs
                  Wing lift =      0 lbs
                  Wing drag =      0 lbs
       Horizontal tail lift =   1541 lbs
       Horizontal tail drag =    126 lbs
    Vertical tail side force =      0 lbs
          Vertical tail drag =     14 lbs
            Auxiliary thrust =      0 lbs
              Tip path angle =   9.06 degs
          Rotor coning angle =   4.16 degs
Location of mean thrust (r/R) =   0.60
  Collective pitch at .7 r/R =   8.83 degs
```

1st lat cyclic term-A1 (deg) =	1.46	
1st long cyclic term-B1 (deg) =	-4.25	
solidity =	0.082	
Disk loading =	0.00	lbs/ft^2
Figure of Merit =	0.00	
CT/sigma =	0.071	
CQ/sigma =	0.0049	
CH/sigma =	0.0016	
Tip mach of the adv. blade =	0.796	
Advance ratio =	0.270	
Rotor thrust required (TPP) =	15147	lbs
Rotor power required =	1338	h.p.
Rotor torque =	27802	ft-lbs

APPENDIX C. JANRAD 98 INPUT/OUTPUT

The following is the actual h60.prf file from JANRAD 98 used to compare with JANRAD 3.1 output.

```
                *** RESULTS ***

            Forward velocity =    115 kts
                 Temperature =     66 degs F
           Pressure altitude =   1000 ft
                Gross weight =  16500 lbs
            Number of blades =      4
                Rotor radius =  26.83 ft
           Blade mean chord =   1.73 ft
                 Blade twist = -18.00 degs
     Blade lift curve slope =   5.73
                Blade weight = 175.00 lbs
        Rotational velocity =  26.47 rads/sec
          Blade grip length =   3.00 ft
                Hinge offset =   1.26 ft
  Equivalent flat plate area =  45.00 ft^2
    Vertical projected area = 227.50 ft^2
                  Wing area =   0.00 ft^2
                  Wing span =   0.00 ft
                    Wing CL =   0.00
                   Wing CDo = 0.0000
     Wing efficiency factor =   0.00
       Horizontal tail area =  45.00 ft^2
       Horizontal tail span =  14.38 ft
         Horizontal tail CL =   0.80
        Horizontal tail CDo = 0.0100
         Vertical tail area =  32.30 ft^2
         Vertical tail span =   8.17 ft
           Vertical tail CL =   0.00
          Vertical tail CDo = 0.0100
              Fuselage drag =   1927 lbs
                 Rotor drag =    336 lbs
                  Wing lift =      0 lbs
                  Wing drag =      0 lbs
       Horizontal tail lift =   1541 lbs
       Horizontal tail drag =    126 lbs
  Vertical tail side force =      0 lbs
        Vertical tail drag =     14 lbs
          Auxiliary thrust =      0 lbs
            Tip path angle =   9.06 degs
        Rotor coning angle =   4.16 degs
Location of mean thrust (r/R) =   0.60
  Collective pitch at .7 r/R =   8.83 degs
```

1st lat cyclic term-A1 (deg) =	1.46	
1st long cyclic term-B1 (deg) =	-4.25	
solidity =	0.082	
Disk loading =	0.00	lbs/ft^2
Figure of Merit =	0.00	
CT/sigma =	0.071	
CQ/sigma =	0.0049	
CH/sigma =	0.0016	
Tip mach of the adv. blade =	0.796	
Advance ratio =	0.270	
Rotor thrust required (TPP) =	15147	lbs
Rotor power required =	1338	h.p.
Rotor torque =	27802	ft-lbs

APPENDIX D. JANRAD98.M

This file launches JANRAD 98 and its Graphical User Interface. It is called by typing janrad98 at the MATLAB command line.

```
function janrad98()

%
%                    JANRAD98.M

%   Joint Army Navy Rotorcraft Analysis and Design
%                      (JANRAD)
%                    Version 4.0
%                     March 1998

%     Version 1.0 Designers
%     MAJ Bob Nicholson
%     MAJ Walter Wirth

%     Version 2.0 Update Designer
%     LT Dale Feddersen

%     Version 3.0 update Designer
%     LT Dave Eccles

%     Version 3.1 update Designer
%     LT Dan Hiatt

%     Version 4.0 update Designer
%     LCDR. Chris F. Lapacik

% Version 4.0 added the Graphical User Interface.  The GUI allows the
user
% to create, run, save and print files with less effort and greater
speed.
% The basic performance calculation routines remain essentially the
same as
% version 3.1.  However, the input, output and file structure were
modified
% extensively.  Sikorsky UH-60A airfoil data was also added.

% Version 3.1 adds time varying tip loss and the corrected dynamics
module.
% The dynamics module provides Southwell plots and rotor blade
response in
% flap and lag motion.  Also included is rotor flapping stability
determination
% by Floquet analysis.
```

```
%  Version 2.0 corrected minor bugs in ver 1.0 and incorporated
%  Wheatley's Eqn.  Additionally the user may now input a
%  tapered rotor blade.  Finally a menu was created for
%  ease of performing various iterations and then saving that
%  calculated data for later use/manipulation.


%  This program is an interactive preliminary design tool
%  developed to aid the design student in determination of
%  initial rotorcraft configurations and in the calculation
%  of performance, stability and control, and other parameters.
%  The program will work for conventional or compound rotorcraft.
%  It will provide accurate data for airspeeds less than 10
%  knots and greater than or equal to 50 knots.


load janrad98

global H_JAN H_EREF H_CNF

H_JAN = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','JANRAD 98', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.0625 0.954688 0.8625], ...
   'Tag','Fig1');
b = uicontrol('Parent',H_JAN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',16, ...
   'FontWeight','bold', ...
   'Position',[0.0604396 0.757143 0.434066 0.145714], ...
   'String','Welcome to JANRAD 98', ...
   'Style','text', ...
   'Tag','StaticText1');
b = uicontrol('Parent',H_JAN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',10, ...
   'FontWeight','bold', ...
   'Position',[0.0758597 0.617284 0.403226 0.0925926], ...
   'String','Joint Army/Navy Rotor Analysis and Design', ...
   'Style','text', ...
   'Tag','StaticText7');
b = uicontrol('Parent',H_JAN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0766129 0.395062 0.403226 0.0617284], ...
   'String','Department of Aeronautics & Astronautics', ...
   'FontSize',10, ...
   'FontWeight','bold', ...
```

```
    'Style','text', ...
    'Tag','StaticText6');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0766129 0.333333 0.403226 0.0617284], ...
    'String','Naval Postgraduate School', ...
    'FontSize',10, ...
    'FontWeight','bold', ...
    'Style','text', ...
    'Tag','StaticText5');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0766129 0.271605 0.403226 0.0617284], ...
    'String','Monterey,CA 93940', ...
    'FontSize',10, ...
    'FontWeight','bold', ...
    'Style','text', ...
    'Tag','StaticText4');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0766129 0.17284 0.403226 0.0771605], ...
    'String','March 25, 1998', ...
    'FontSize',10, ...
    'FontWeight','bold', ...
    'Style','text', ...
    'Tag','StaticText3');
H_EREF = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'Callback','janrad98_fcn eref', ...
    'Position',[0.587992 0.655172 0.275362 0.103448], ...
    'String','Edit/Run Existing File', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton1', ...
    'Value',1);
H_CNF = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'Callback','janrad98_fcn cnf', ...
    'Position',[0.587992 0.514368 0.275362 0.106322], ...
    'String','Create New File', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton2');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'Callback','janrad98_fcn stop', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.587992 0.295977 0.275362 0.112069], ...
    'String','Quit JANRAD 98', ...
    'Tag','Pushbutton1');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
```

```
    'Callback','janrad98_fcn cont', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.590062 0.149425 0.273292 0.112069], ...
    'String','Continue >>', ...
    'Tag','Pushbutton2');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','demi', ...
    'Position',[0.591097 0.793103 0.269151 0.091954], ...
    'String','Select Option and Continue', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.561077 0.445402 0.329193 0.482759], ...
    'Style','frame', ...
    'Tag','Frame1');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0342742 0.570988 0.479839 0.367284], ...
    'Style','frame', ...
    'Tag','Frame2');
b = uicontrol('Parent',H_JAN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0687398 0.149758 0.417349 0.345411], ...
    'Style','frame', ...
    'Tag','Frame3');

assignin('base','H_EREF',H_EREF);
assignin('base','H_CNF',H_CNF);
```

APPENDIX E. JANRAD98_FCN.M

Switchyard Callback function called by the janrad98.m GUI function.

```
function janrad_fcn(Action)

%  Switchyard Callback function for janrad98.m
%  JANRAD 98 VERSION 4.0

global H_EREF H_CNF H_JAN NAME COUNT

cond1=get(H_EREF,'Value');
cond2=get(H_CNF,'Value');

COUNT=0;

if nargin,
   switch Action
   case 'cont',
      if cond1==1
         analysis
         close (H_JAN)
      else,
         NAME=[];
         performance_input
         close (H_JAN)
      end
   case 'stop'
      quit_gui
   case 'eref'
      set(H_EREF,'Value',1)
      set(H_CNF,'Value',0)
   case 'cnf'
      set(H_EREF,'Value',0)
      set(H_CNF,'Value',1)
   end
end
```

APPENDIX F. ANALYSIS.M

This file creates the GUI to select a JANRAD 98 data file and select the analysis method. It is called in the janrad98_fcn.m Switchyard Callback function.

```
function analysis()

%  GUI figure window to Select Type Analysis
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load analysis

global H_P H_SAC H_RD H_ANAL H_LB NAME

H_ANAL = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'Name','Analysis', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.0625 0.954688 0.8625], ...
   'Tag','Fig2');
b = uimenu('Parent',H_ANAL, ...
   'Label','JANRAD Options', ...
   'Tag','uimenu1');
c = uimenu('Parent',b, ...
   'Callback','analysis_fcn quit', ...
   'Label','Quit JANRAD', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','analysis_fcn return',...
   'Label','Return to Begining', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','analysis_fcn delta_input',...
   'Label','Change Input Parameters', ...
   'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
   'Callback','analysis_fcn about',...
```

```
    'Label','About Janrad 98 ...', ...
    'Separator','on',...
    'Tag','Subuimenu1');
b = uicontrol('Parent',H_ANAL, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontWeight','bold', ...
    'Position',[0.106033 0.896175 0.332724 0.0546448], ...
    'String','Type Path to Working Directory', ...
    'Style','text', ...
    'Tag','StaticText2');
H_WORK = uicontrol('Parent',H_ANAL, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.104205 0.822404 0.330896 0.0546448], ...
    'CreateFcn','',...
    'String',pwd,...
    'HorizontalAlignment','left',...
    'Style','edit', ...

'Callback','cd(get(H_WORK,''String''));list=dir(''*.mat'');str={list.na
me};set(H_LB,''str'',str)',...
    'Tag','EditText1');
b = uicontrol('Parent',H_ANAL, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.102377 0.734973 0.340037 0.0628415], ...
    'String','Select Data File', ...
    'Style','text', ...
    'Tag','StaticText3');
H_LB = uicontrol('Parent',H_ANAL, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.101786 0.355742 0.342857 0.369748], ...
    'String','', ...

'CreateFcn','list=dir(''*.mat'');str={list.name};set(gcbo,''str'',str)'
, ...
    'Style','listbox', ...
    'Callback','global
NAME;Value=get(gcbo,''Value'');String=get(gcbo,''String'');NAME=String{
Value};', ...
    'Tag','Listbox1', ...
    'Value',1);
b = uicontrol('Parent',H_ANAL, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.598214 0.815126 0.285714 0.0644258], ...
    'String','Select Type Analysis', ...
    'Style','text', ...
```

```
   'Tag','StaticText1');
H_P = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback','analysis_fcn h_p', ...
   'FontSize',12, ...
   'Position',[0.598214 0.672269 0.285714 0.092437], ...
   'String','Performance', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton1', ...
   'Value',1);
H_SAC = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback','analysis_fcn h_sac', ...
   'FontSize',12, ...
   'Position',[0.598214 0.540616 0.285714 0.092437], ...
   'String','Stability And Control', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton1');
H_RD = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback','analysis_fcn h_rd', ...
   'FontSize',12, ...
   'Position',[0.598214 0.408964 0.285714 0.0952381], ...
   'String','Rotor Dynamics', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton1');
b = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'Callback','analysis_fcn back', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.0982143 0.176 0.210714 0.096], ...
   'String','<< Back', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'Callback','analysis_fcn cnx',...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.391071 0.176 0.208929 0.096], ...
   'String','Cancel', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',H_ANAL, ...
   'Units','normalized', ...
   'Callback','global COUNT;COUNT=0;analysis_fcn cont', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.678571 0.178667 0.208929 0.096], ...
   'String','Continue >>', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',H_ANAL, ...
```

```
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.557143 0.352941 0.355357 0.557423], ...
    'Style','frame', ...
    'Tag','Frame1');

assignin('base','H_P',H_P);
assignin('base','H_SAC',H_SAC);
assignin('base','H_RD',H_RD);
assignin('base','H_WORK',H_WORK);
assignin('base','H_LB',H_LB);
assignin('caller','NAME',NAME)
```

APPENDIX G. ANALYSIS_FCN.M

Switchyard Callback function for the analysis.m GUI function.

```
function analysis_fcn(Action)

%  Switchyard Callback function for analysis.m
%  JANRAD 98 VERSION 4.0

global H_P H_SAC H_RD H_ANAL

cond1=get(H_P,'Value');
cond2=get(H_SAC,'Value');
cond3=get(H_RD,'Value');

if nargin,
   switch Action
   case 'h_p'
      set(H_P,'Value',1)
      set(H_SAC,'Value',0)
      set(H_RD,'Value',0)
   case 'h_sac'
      set(H_P,'Value',0)
      set(H_SAC,'Value',1)
      set(H_RD,'Value',0),
   case 'h_rd'
      set(H_P,'Value',0)
      set(H_SAC,'Value',0)
      set(H_RD,'Value',1)
   case 'cont'
      if cond1==1
         performance_inPut
         close (H_ANAL)
      elseif cond2==1
         stability_and_control
      elseif cond3==1
         rotor_dynamics
      else,
         error('SomeThing is Wrong in Analysis Function')
      end
   case 'cnx'
      analysis
      close (gcf)
   case 'back'
      janrad98
      close (H_ANAL)
   case 'return'
      janrad98
      close all
   case 'quit'
```

```
      quit_gui
   case 'delta_input'
      performance_input
      close (H_ANAL)
   case 'about'
      about_janrad
   end
end
```

APPENDIX H. PERFORMANCE_INPUT.M

This file creates the GUI that displays the 36 input parameters as loaded from a previously saved file or created new by user. It is called in the analysis_fcn.m Switchyard Callback function.

```
function performance_input()

%  GUI window to display and/or edit input values.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load performance_input

global COUNT NAME H_PERF_IN S_USER_INPUT S_PERF_INPUT

switch COUNT
case 0
   if ~isempty(NAME)
      eval(['load ',NAME])
      unstructure1
   else
      load create_new
      structure
   end
case 1
   unstructure1
end


H_PERF_IN = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'CreateFcn','',...
   'Colormap',mat0, ...
   'Name','Performance Input Parameters', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.0625 0.954688 0.8625], ...
   'Tag','Fig2');
```

```
h_opt = uimenu('Parent',H_PERF_IN, ...
   'Label','JANRAD Options', ...
   'Tag','uimenu1');
c = uimenu('Parent',h_opt, ...
   'Callback','performance_input_fcn quit', ...
   'Label','Quit JANRAD', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',h_opt, ...
   'Callback','performance_input_fcn return',...
   'Label','Return to Begining', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',h_opt, ...
   'Callback','performance_input_fcn delta_input',...
   'Label','Change Input Parameters', ...
   'Enable','off',...
   'Tag','Subuimenu1');
c = uimenu('Parent',h_opt, ...
   'Callback','performance_input_fcn about',...
   'Label','About Janrad 98 ...', ...
   'Separator','on',...
   'Tag','Subuimenu1');

d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.92029 0.175123 0.0483092], ...
   'String','Pressure Altitude (ft)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.922705 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',PA,...

'Callback','PA=get(gcbo,''String'');S_USER_INPUT.PA=str2num(PA);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.855072 0.175123 0.0483092], ...
   'String','Temperature (deg F)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.857488 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',temp,...
```

```
'Callback','temp=get(gcbo,''String'');S_USER_INPUT.temp=str2num(temp);'
,...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.792271 0.175123 0.0483092], ...
   'String','Airspeed (kts)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.792271 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',Vinf,...

'Callback','Vinf=get(gcbo,''String'');S_USER_INPUT.Vinf=str2num(Vinf);'
,...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.727053 0.175123 0.0483092], ...
   'String','Gross Wt. (lbs.)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.729469 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',GW,...

'Callback','GW=get(gcbo,''String'');S_USER_INPUT.GW=str2num(GW);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.661836 0.175123 0.0483092], ...
   'String','Rotor Vel. (rad/sec)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.664251 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',omega,...

'Callback','omega=get(gcbo,''String'');S_USER_INPUT.omega=str2num(omega
);',...
   'Tag','EditText1');
```

```
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0212766 0.596618 0.173486 0.0483092], ...
    'String','No. Azmith Sectors', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.217676 0.599034 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',naz,...

'Callback','naz=get(gcbo,''String'');S_USER_INPUT.naz=str2num(naz);',..
.
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0212766 0.533816 0.175123 0.0483092], ...
    'String','Coll Pitch @ .7 r/R', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.217676 0.536232 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',thetao,...

'Callback','thetao=get(gcbo,''String'');S_USER_INPUT.thetao=str2num(the
tao);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0212766 0.468599 0.173486 0.0483092], ...
    'String','Wing Area (ft^2)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.217676 0.471014 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',Swing,...

'Callback','Swing=get(gcbo,''String'');S_USER_INPUT.Swing=str2num(Swing
);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
   'Position',[0.0212766 0.403382 0.173486 0.0483092], ...
   'String','Wing Span (ft)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.405797 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',bwing,...

'Callback','bwing=get(gcbo,''String'');S_USER_INPUT.bwing=str2num(bwing
);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.34058 0.173486 0.0483092], ...
   'String','Expected Wing CL', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.342995 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CLwing,...

'Callback','CLwing=get(gcbo,''String'');S_USER_INPUT.CLwing=str2num(CLw
ing);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.275362 0.173486 0.0483092], ...
   'String','Wing CDo', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.217676 0.277778 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CDowing,...

'Callback','CDowing=get(gcbo,''String'');S_USER_INPUT.CDowing=str2num(C
Dowing);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0212766 0.210145 0.173486 0.0483092], ...
   'String','Wing Eff. Factor - e', ...
   'Style','text', ...
```

```
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.217676 0.21256 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',ewing,...

'Callback','ewing=get(gcbo,''String'');S_USER_INPUT.ewing=str2num(ewing
);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.922705 0.173486 0.0483092], ...
    'String','Blade Airfoil Type', ...
    'Style','text', ...
    'Tag','StaticText2');
pop = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.549918 0.922705 0.125 0.0483092], ...
    'String','0012|HH-02|VR-12|SC1094r8|SC1095r8', ...
    'Style','popupmenu', ...
    'Tag','PopupMenu1', ...
    'Value',afoil, ...
    'Callback','afoil=get(gcbo,''Value'');S_USER_INPUT.afoil=afoil;');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.855072 0.173486 0.0483092], ...
    'String','Blade Lift Curve Slope', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.857488 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',a,...
    'Callback','a=get(gcbo,''String'');S_USER_INPUT.a=str2num(a);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.794686 0.175123 0.0458937], ...
    'String','No. Blades', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.792271 0.0981997 0.0483092], ...
    'Style','edit', ...
```

```
    'String',b,...
    'Callback','b=get(gcbo,''String'');S_USER_INPUT.b=str2num(b);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.731884 0.175123 0.0483092], ...
    'String','Blade Radius (ft.)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.729469 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',R,...
    'Callback','R=get(gcbo,''String'');S_USER_INPUT.R=str2num(R);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.664251 0.175123 0.0483092], ...
    'String','Hinge Offset (ft.)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.664251 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',e,...
    'Callback','e=get(gcbo,''String'');S_USER_INPUT.e=str2num(e);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.601449 0.175123 0.0483092], ...
    'String','Non-Aero Part (ft.)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.599034 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',grip,...

'Callback','grip=get(gcbo,''String'');S_USER_INPUT.grip=str2num(grip);'
,...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.538647 0.175123 0.0483092], ...
```

```
    'String','Blade Root Chd (ft.)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.536232 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',rchord,...

'Callback','rchord=get(gcbo,''String'');S_USER_INPUT.rchord=str2num(rch
ord);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.471014 0.173486 0.0483092], ...
    'String','Blade Taper Ratio', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.471014 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',tr,...

'Callback','tr=get(gcbo,''String'');S_USER_INPUT.tr=str2num(tr);',...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.405797 0.175123 0.0483092], ...
    'String','Taper Starts @ (r/R)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.549918 0.405797 0.0981997 0.0483092], ...
    'Style','edit', ...
    'String',trst,...

'Callback','trst=get(gcbo,''String'');S_USER_INPUT.trst=str2num(trst);'
,...
    'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.358429 0.342995 0.175123 0.0483092], ...
    'String','Blade Twist (deg)', ...
    'Style','text', ...
    'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
```

```
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.549918 0.342995 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',twist,...

'Callback','twist=get(gcbo,''String'');S_USER_INPUT.twist=str2num(twist
);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.358429 0.280193 0.175123 0.0483092], ...
   'String','Blade Wt-Aero (lbs.)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.549918 0.277778 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',wblade,...

'Callback','wblade=get(gcbo,''String'');S_USER_INPUT.wblade=str2num(wbl
ade);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.358429 0.21256 0.175123 0.0483092], ...
   'String','No. Blade Elements', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.549918 0.21256 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',nbe,...

'Callback','nbe=get(gcbo,''String'');S_USER_INPUT.nbe=str2num(nbe);',..
.
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.916264 0.175123 0.0483092], ...
   'String','Auxillary Thrust (lbs)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.917874 0.0981997 0.0483092], ...
```

```
   'Style','edit', ...
   'String',Taux,...

'Callback','Taux=get(gcbo,''String'');S_USER_INPUT.Taux=str2num(Taux);'
,...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',6, ...
   'Position',[0.711948 0.851852 0.173486 0.0483092], ...
   'String','Flat Plate Area (ft^2)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.852657 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',Afh,...

'Callback','Afh=get(gcbo,''String'');S_USER_INPUT.Afh=str2num(Afh);',..
.
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.78744 0.173486 0.0483092], ...
   'String','Vert. Proj Area (ft^2)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.789855 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',Afv,...

'Callback','Afv=get(gcbo,''String'');S_USER_INPUT.Afv=str2num(Afv);',..
.
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.723027 0.174577 0.0483092], ...
   'String','Vert. Tail Area (ft^2)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.724638 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',Svert,...
```

```
'Callback','Svert=get(gcbo,''String'');S_USER_INPUT.Svert=str2num(Svert
);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.658615 0.174577 0.0483092], ...
   'String','Vert. Tail Span (ft)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.65942 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',bvert,...

'Callback','bvert=get(gcbo,''String'');S_USER_INPUT.bvert=str2num(bvert
);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.594203 0.174577 0.0483092], ...
   'String','Vert. Tail CL', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.596618 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CLvert,...

'Callback','CLvert=get(gcbo,''String'');S_USER_INPUT.CLvert=str2num(CLv
ert);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.529791 0.174577 0.0483092], ...
   'String','Vert. Tail CDo', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.531401 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CDovert,...

'Callback','CDovert=get(gcbo,''String'');S_USER_INPUT.CDovert=str2num(C
Dovert);',...
```

```
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.465378 0.174577 0.0483092], ...
   'String','Horiz. Tail Area (ft^2)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.466184 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',Shoriz,...

'Callback','Shoriz=get(gcbo,''String'');S_USER_INPUT.Shoriz=str2num(Sho
riz);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.400966 0.174577 0.0483092], ...
   'String','Horiz. Tail Span (ft)', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.403382 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',bhoriz,...

'Callback','bhoriz=get(gcbo,''String'');S_USER_INPUT.bhoriz=str2num(bho
riz);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.336554 0.174577 0.0483092], ...
   'String','Horiz. Tail CL', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.338164 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CLhoriz,...

'Callback','CLhoriz=get(gcbo,''String'');S_USER_INPUT.CLhoriz=str2num(C
Lhoriz);',...
   'Tag','EditText1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
```

```
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.272141 0.174577 0.0483092], ...
   'String','Horiz. Tail CDo', ...
   'Style','text', ...
   'Tag','StaticText2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[1 1 1], ...
   'Position',[0.90671 0.272947 0.0981997 0.0483092], ...
   'Style','edit', ...
   'String',CDohoriz,...

'Callback','CDohoriz=get(gcbo,''String'');S_USER_INPUT.CDohoriz=str2num
(CDohoriz);',...
   'Tag','EditText1');
H_DISK = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.711948 0.207729 0.297872 0.0483092], ...
   'String','Horiz. Tail Under Main Rotor Disk', ...
   'Style','checkbox', ...
   'Tag','Checkbox1', ...
   'Value',0, ... .
   'CreateFcn','taildisk=2;', ...
   'Callback',[...
      'if get(H_DISK, ''Value'')==1,'...
      'taildisk=1;'...
      'else,'...
      'taildisk=2;'...
      'end,'...
      'S_USER_INPUT.taildisk=taildisk;']);

d = uicontrol('Parent',H_PERF_IN, ...
   'Units','normalized', ...
   'Callback','performance_input_fcn back', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.111293 0.0555556 0.163666 0.0805153], ...
   'String','<< Back', ...
   'Tag','Pushbutton1');
d = uicontrol('Parent',H_PERF_IN, ...
   'Callback','performance_input_fcn print',...
   'Units','normalized', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.318603 0.0555556 0.163666 0.0805153], ...
   'String','Print Screen', ...
   'Tag','Pushbutton2');
d = uicontrol('Parent',H_PERF_IN, ...
   'Callback','performance_input_fcn cnx',...
   'Units','normalized', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.525914 0.0555556 0.163666 0.0805153], ...
```

```
    'String','Cancel', ...
    'Tag','Pushbutton3');
d = uicontrol('Parent',H_PERF_IN, ...
    'Units','normalized', ...
    'Callback','global REGIME PICK
S_PERF_INPUT,PICK=0;REGIME=0;S_PERF_INPUT=S_USER_INPUT;performance_inpu
t_fcn cont', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.733224 0.0555556 0.163666 0.0805153], ...
    'String','Continue >>', ...
    'Tag','Pushbutton4');

assignin('base','H_DISK',H_DISK);
assignin('base','S_USER_INPUT',S_USER_INPUT);
assignin('base','S_PERF_INPUT',S_PERF_INPUT);
```

APPENDIX I. PERFORMANCE_INPUT_FCN.M

Switchyard Callback function for the performance_input.m GUI function.

```
function performance_input_fcn(Action)

%  Switchyard Callback function for performance_input.m
%  JANRAD 98 VERSION 4.0

global  H_PERF_IN H_IT_METH S_PERF_INPUT

if nargin,
   switch Action
   case 'cont'
      if isempty(getfield(S_PERF_INPUT,'PA')|...
            getfield(S_PERF_INPUT,'temp')|...
            getfield(S_PERF_INPUT,'Vinf')|...
            getfield(S_PERF_INPUT,'GW')|...
            getfield(S_PERF_INPUT,'omega')|...
            getfield(S_PERF_INPUT,'naz')|...
            getfield(S_PERF_INPUT,'thetao')|...
            getfield(S_PERF_INPUT,'Swing')|...
            getfield(S_PERF_INPUT,'bwing')|...
            getfield(S_PERF_INPUT,'CLwing')|...
            getfield(S_PERF_INPUT,'CDowing')|...
            getfield(S_PERF_INPUT,'ewing')|...
            getfield(S_PERF_INPUT,'afoil')|...
            getfield(S_PERF_INPUT,'a')|...
            getfield(S_PERF_INPUT,'b')|...
            getfield(S_PERF_INPUT,'R')|...
            getfield(S_PERF_INPUT,'e')|...
            getfield(S_PERF_INPUT,'grip')|...
            getfield(S_PERF_INPUT,'rchord')|...
            getfield(S_PERF_INPUT,'tr')|...
            getfield(S_PERF_INPUT,'trst')|...
            getfield(S_PERF_INPUT,'twist')|...
            getfield(S_PERF_INPUT,'wblade')|...
            getfield(S_PERF_INPUT,'nbe')|...
            getfield(S_PERF_INPUT,'Taux')|...
            getfield(S_PERF_INPUT,'Afh')|...
            getfield(S_PERF_INPUT,'Afv')|...
            getfield(S_PERF_INPUT,'Svert')|...
            getfield(S_PERF_INPUT,'bvert')|...
            getfield(S_PERF_INPUT,'CLvert')|...
            getfield(S_PERF_INPUT,'CDovert')|...
            getfield(S_PERF_INPUT,'Shoriz')|...
            getfield(S_PERF_INPUT,'bhoriz')|...
            getfield(S_PERF_INPUT,'CLhoriz')|...
            getfield(S_PERF_INPUT,'CDohoriz'));
         empty_boxes
```

```
      else
         iteration_method
         close (H_PERF_IN)
      end
   case 'cnx'
      performance_input
      close (gcf)
   case 'back'
      analysis
      close (H_PERF_IN)
   case 'print'
      set(gcf,'PaperOrientation','landscape')
      set(gcf,'PaperPosition',[.5 .5 10 7.5])
      print -dwinc
   case 'return'
      janrad98
      close all
   case 'quit'
      quit_gui
   case 'about'
      about_janrad
   end
end
```

APPENDIX J. PERFORMANCE_OUTPUT.M

This file creates the GUI to display the calculated results from a previously saved input file or newly created user input. It is call in Perf.m.

```
function performance_output()

%  GUI window to Display Janrad perfromance output.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load performance_output

global COUNT H_PERF_OUT S_PERF_OUTPUT S_USER_INPUT ...
   H_datain H_dataout H_vecdata H_check1 H_check2 H_check3

COUNT=1;

H_PERF_OUT = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'Name','Performance Output', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.05625 0.954688 0.86875], ...
   'Tag','Fig1');
b = uimenu('Parent',H_PERF_OUT, ...
   'Label','JANRAD Options', ...
   'Tag','uimenu1');
c = uimenu('Parent',b, ...
   'Callback','performance_output_fcn quit', ...
   'Label','Quit JANRAD', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','performance_output_fcn return',...
   'Label','Return to Begining', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','performance_output_fcn delta_input',...
   'Label','Change Input Parameters', ...
```

```
    'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
    'Callback','performance_output_fcn about',...
    'Label','About Janrad 98 ...', ...
    'Separator','on',...
    'Tag','Subuimenu1');

b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.925659 0.327332 0.0383693], ...
    'String','Fuselage Drag (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.925659 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Dfuse, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.872902 0.327332 0.0383693], ...
    'String','Rotor Drag (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.872902 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Hrotor, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.817746 0.327332 0.0383693], ...
    'String','Wing Lift (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.817746 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Lwing, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
```

```
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.76259 0.327332 0.0383693], ...
    'String','Wing Drag (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.76259 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Dwing, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.709832 0.327332 0.0383693], ...
    'String','Horizontal Tail Lift (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.707434 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Lhoriz, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.654676 0.327332 0.0383693], ...
    'String','Horizontal Tail Drag (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.654676 0.0981997 0.0383693], ...
    'Style','text', ...
    'String',S_PERF_OUTPUT.Dhoriz', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.59952 0.327332 0.0383693], ...
    'String','Vertical Tail Lift (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
    'Position',[0.376432 0.59952 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Lvert, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.546763 0.327332 0.0383693], ...
    'String','Vertical Tail Drag (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.546763 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Dvert, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.491607 0.327332 0.0383693], ...
    'String','Tip Path Angle (deg)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.491607 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.alphaT, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.436451 0.327332 0.0383693], ...
    'String','Rotor Coning Angle (deg)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.436451 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.betao, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.383693 0.327332 0.0383693], ...
```

```
    'String','Location of Main Thrust (r/R)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.383693 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.rT2, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.328537 0.327332 0.0383693], ...
    'String','1st Lat. Cyclic Term - A1', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.328537 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.theta1c, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.0310966 0.273381 0.327332 0.0383693], ...
    'String','1st Long. Cyclic Term - B1', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.376432 0.273381 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.theta1s, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.923261 0.327332 0.0383693], ...
    'String','Collective Pitch @ .7 r/R (deg)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.925659 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.thetao, ...
    'Style','text', ...
```

```
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.870504 0.327332 0.0383693], ...
    'String','Solidity (sigma)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.872902 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.solidity, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.815348 0.327332 0.0383693], ...
    'String','Disk Loading (lbs. /ft^2)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.817746 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.DL, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.760192 0.327332 0.0383693], ...
    'String','Figure of Merit', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.76259 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.FM, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.707434 0.327332 0.0383693], ...
    'String','CT/Sigma', ...
    'Style','text', ...
    'Tag','StaticText1');
```

```
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.709832 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.CT_sig, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.652278 0.327332 0.0383693], ...
    'String','CQ/Sigma', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.654676 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.CQ_sig, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.597122 0.327332 0.0383693], ...
    'String','CH/Sigma', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.59952 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.CH_sig, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.544365 0.327332 0.0383693], ...
    'String','Tip Mach No. of Advancing Blade', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.546763 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Machtip, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
```

```
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.489209 0.327332 0.0383693], ...
    'String','Advance Ratio', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.491607 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.mu, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.434053 0.327332 0.0383693], ...
    'String','Rotor Thrust Required - TPP (lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.436451 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.T, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.381295 0.327332 0.0383693], ...
    'String','Rotor Power Required (hp)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.383693 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Protor, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.326139 0.327332 0.0383693], ...
    'String','Rotor Torque (ft.-lbs.)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
    'Position',[0.859247 0.328537 0.0981997 0.0383693], ...
    'String',S_PERF_OUTPUT.Qrotor, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',10, ...
    'Position',[0.512275 0.270983 0.327332 0.0383693], ...
    'String','Auxilliary Thrust (lbs)', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.859247 0.273381 0.0981997 0.0383693], ...
    'String',S_USER_INPUT.Taux, ...
    'Style','text', ...
    'Tag','StaticText1');
H_check1 = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0310966 0.177458 0.266776 0.0479616], ...
    'String','Save Input Data as ....', ...
    'Style','checkbox', ...
    'Tag','Checkbox1');
H_datain = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.302782 0.177458 0.0981997 0.0479616], ...
    'FontSize',12, ...
    'Style','edit', ...
    'String','',...
    'Callback',[...
       'set(gcbo,''String'',get(gcbo,''String''));','...
       'set(H_dataout,''String'',get(H_datain,''String''));','...
       'set(H_vecdata,''String'',get(H_datain,''String''));','...
       'set(H_check1,''Value'',1);','...
       'set(H_check2,''Value'',1);','...
       'set(H_check3,''Value'',1);'],...
    'HorizontalAlignment','right',...
    'Tag','EditText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.405892 0.179856 0.0981997 0.0479616], ...
    'Style','text', ...
    'String','.mat',...
    'HorizontalAlignment','left',...
    'Tag','StaticText2');
H_check2 = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
    'Position',[0.0310966 0.117506 0.266776 0.0479616], ...
    'String','Save Output Data as ....', ...
    'Style','checkbox', ...
    'Tag','Checkbox1');
H_dataout = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.302782 0.119904 0.0981997 0.0479616], ...
    'FontSize',12, ...
    'String','',...
    'Style','text', ...
    'HorizontalAlignment','right',...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.405892 0.122302 0.0981997 0.0479616], ...
    'String','.prf', ...
    'HorizontalAlignment','left',...
    'Style','text', ...
    'Tag','StaticText2');
H_check3 = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0310966 0.059952 0.266776 0.0479616], ...
    'String','Save Matrix & Vector Data as ....', ...
    'Style','checkbox', ...
    'Tag','Checkbox1');
H_vecdata = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.302782 0.0623501 0.0981997 0.0479616], ...
    'FontSize',12, ...
    'Style','text', ...
    'HorizontalAlignment','right',...
    'Tag','StaticText1');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.405892 0.0647482 0.0981997 0.0479616], ...
    'String','_p.mat', ...
    'HorizontalAlignment','left',...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.572831 0.146283 0.140753 0.0815348], ...
    'String','<< Back', ...
    'Tag','Pushbutton1',...
    'Callback','performance_output_fcn back');
```

```
b = uicontrol('Parent',H_PERF_OUT, ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.749591 0.146283 0.140753 0.0815348], ...
    'String','Options >>', ...
    'Tag','Pushbutton1',...
    'Callback','performance_output_fcn opt');
b = uicontrol('Parent',H_PERF_OUT, ...
    'Callback','performance_output_fcn print',...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.574468 0.059952 0.314239 0.0527578], ...
    'String','Print Screen', ...
    'Tag','Pushbutton1');

assignin('base','H_datain',H_datain);
assignin('base','H_dataout',H_dataout);
assignin('base','H_vecdata',H_vecdata);
assignin('base','H_check1',H_check1);
assignin('base','H_check2',H_check2);
assignin('base','H_check3',H_check3);
```

APPENDIX K. PERFORMANCE_OUTPUT_FCN.M

Switchyard Callback function for the performacne_output.m GUI function.

```
function performance_output_fcn(Action)

%  Switchyard Callback for performance_output.m
%  JANRAD 98 VERSION 4.0

global H_PERF_OUT S_USER_INPUT S_PERF_INPUT S_PERF_OUTPUT S_MATR_VEC...
   H_datain H_dataout H_vecdata H_check1 H_check2 H_check3 ...
   H_outputfile H_vecfile H_inputfile

if nargin
   switch Action
   case 'back'
      close (H_PERF_OUT)
      S_PERF_INPUT=S_USER_INPUT;
      iteration_method
   case 'opt'
      if get(H_check1,'Value')==1
         S_USER_INPUT=S_PERF_INPUT;
         S_USER_INPUT.Vinf=S_USER_INPUT.Vinf/1.68894444;
         S_USER_INPUT.thetao=S_USER_INPUT.thetao*57.3;
         S_USER_INPUT.twist=-S_USER_INPUT.twist*57.3;
         filename1=get(H_datain,'String');
         eval(['save ',filename1,' S_USER_INPUT'])
      end
      if get(H_check2,'Value')==1
         filename1=get(H_datain,'String');
         eval(['!copy print_temp1 ', filename1,'.prf'])
      end
      if get(H_check3,'Value')==1
         unstructure3
         filename2=[filename1 '_p'];
         eval(['save ',filename2,' r psi vi theta betat alpha Tpsi Mpsi
DMpsi dT dM dD cblade CL CD']);
      end
      options
      set(H_inputfile,'String',[filename1,'.mat'])
      set(H_outputfile,'String',[get(H_dataout,'String'),'.prf'])
      set(H_vecfile,'String',[get(H_vecdata,'String'),'_p.mat'])
      close (H_PERF_OUT)
   case 'print'
      set(gcf,'PaperOrientation','landscape')
      set(gcf,'PaperPosition',[.5 .5 10 7.5])
      print -dwinc
   case 'return'
      close all
      janrad98
```

```
    case 'delta_input'
       close (H_PERF_OUT)
       performance_input
    case 'quit'
       quit_gui
    case 'about'
       about_janrad
    end
end
```

APPENDIX L. ITERATION_METHOD.M

This file creates GUI to select iteration method and display the status of JANRAD 98 computations. Status comments are set in Trim.m and Perf.m. When computations are complete, this window is closed in Perf.m.

```
function iteration_method()

%  GUI window to select iteration method,  start computational
routines,
%  and display clock and performance method status.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load iteration_method

global H_IT_METH H_NI H_AS H_AL H_GW H_BT H_BTR H_SOT H_WSA ...
   H_STATUS H_STATUS1 H_STATUS2 ...
   H_GO H_RUPT H_BK H_RES H_MEN ...
   COUNT S_USER_INPUT S_PERF_INPUT REGIME PICK ...

COUNT=1;
S_USER_INPUT=S_PERF_INPUT;

H_IT_METH = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'Name','Iteration Method', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.0625 0.954688 0.8625], ...
   'Tag','Fig1');
H_MEN = uimenu('Parent',H_IT_METH, ...
   'Label','JANRAD Options', ...
   'Tag','uimenu1');
c = uimenu('Parent',H_MEN, ...
   'Callback','iteration_method_fcn quit', ...
   'Label','Quit JANRAD', ...
   'Tag','JANRAD OptionsSubuimenu1');
```

```
c = uimenu('Parent',H_MEN, ...
   'Callback','iteration_method_fcn return',...
   'Label','Return to Begining', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',H_MEN, ...
   'Callback','iteration_method_fcn delta_input',...
   'Label','Change Input Parameters', ...
   'Tag','Subuimenu1');
c = uimenu('Parent',H_MEN, ...
   'Callback','iteration_method_fcn about',...
   'Label','About Janrad 98 ...', ...
   'Separator','on',...
   'Tag','Subuimenu1');
f = uicontrol('Parent',H_IT_METH, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.0785714 0.864 0.333333 0.0533333], ...
   'String','Choose Iteration Method', ...
   'Style','text', ...
   'Tag','StaticText1');
H_NI = uicontrol('Parent',H_IT_METH, ...
   'Callback','iteration_method_fcn h_ni',...
   'Value',1,...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'Position',[0.0785714 0.768 0.332143 0.0533333], ...
   'String','No Iteration', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton1');
H_AS = uicontrol('Parent',H_IT_METH, ...
   'Callback','iteration_method_fcn h_as',...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'Position',[0.0785714 0.685333 0.332143 0.0533333], ...
   'String','Airspeed', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton2');
H_AL = uicontrol('Parent',H_IT_METH, ...
   'Callback','iteration_method_fcn h_al',...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'Position',[0.0785714 0.605333 0.332143 0.0533333], ...
   'String','Altitude', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton3');
H_GW = uicontrol('Parent',H_IT_METH, ...
   'Callback','iteration_method_fcn h_gw',...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
    'FontSize',12, ...
    'Position',[0.0785714 0.522667 0.332143 0.0533333], ...
    'String','Gross Weight', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton4');
H_BT = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn h_bt',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.0785714 0.44 0.332143 0.0533333], ...
    'String','Blade Twist', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton5');
H_BTR = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn h_btr',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.0785714 0.36 0.332143 0.0533333], ...
    'String','Blade Taper Ratio', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton6');
H_SOT = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn h_sot',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.0785714 0.277333 0.332143 0.0533333], ...
    'String','Start of Taper', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton7');
H_WSA = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn h_wsa',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.0785714 0.205333 0.332143 0.0533333], ...
    'String','Wing Span Area', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton8');
f = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.482143 0.866667 0.439286 0.0533333], ...
    'String','Analysis Status Box', ...
    'Style','text', ...
    'Tag','StaticText1');
H_STATUS = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.482143 0.186667 0.439286 0.64], ...
```

```
    'Style','text', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'HorizontalAlignment','center',...
    'String','',...
    'Tag','StaticText2');
H_STATUS1 = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.486183 0.411764 0.436029 0.217195], ...
    'Style','text', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'HorizontalAlignment','center',...
    'String','',...
    'Tag','StaticText3');
H_STATUS2 = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.488229 0.191554 0.433981 0.205128], ...
    'Style','text', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'HorizontalAlignment','center',...
    'String','',...
    'Tag','StaticText4');
H_BK = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'Callback','iteration_method_fcn back', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.0767857 0.064 0.178571 0.072], ...
    'String','<< Back', ...
    'Tag','Pushbutton1');
H_GO = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'Callback','global PERF_OUTPUT;REGIME=0;iteration_method_fcn anal',
...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.301786 0.0613333 0.178571 0.072], ...
    'String','Analyze', ...
    'Tag','Pushbutton2');
H_RUPT = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn interrupt',...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.528571 0.0613333 0.178571 0.072], ...
    'String','Interrupt', ...
    'Enable','off',...
    'Tag','Pushbutton3');
H_RES = uicontrol('Parent',H_IT_METH, ...
    'Callback','iteration_method_fcn resume',...
```

```
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.755357 0.0613333 0.178571 0.072], ...
    'String','Resume', ...
    'Enable','off',...
    'Tag','Pushbutton4');
f = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'Position',[0.0678571 0.176 0.355357 0.770667], ...
    'Style','frame', ...
    'Tag','Frame1');
f = uicontrol('Parent',H_IT_METH, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.476786 0.176 0.45 0.768], ...
    'Style','frame', ...
    'Tag','Frame2');

assignin('base','H_NI',H_NI);
assignin('base','H_AS',H_AS);
assignin('base','H_AL',H_AL);
assignin('base','H_GW',H_GW);
assignin('base','H_BT',H_BT);
assignin('base','H_BTR',H_BTR);
assignin('base','H_SOT',H_SOT);
assignin('base','H_WSA',H_WSA);
assignin('base','H_GO',H_GO);
assignin('base','H_RUPT',H_RUPT);
assignin('base','H_BK',H_BK);
assignin('base','H_RES',H_RES);
assignin('base','H_MEN',H_MEN);
```

APPENDIX M. ITERATION_METHOD_FCN.M

Switchyard Callback function for the iteration_method.m GUI function.

```
function iteration_method_fcn(Action)

%  Switchyard Callback for iteration_method.m
%  JANRAD 98 VERSION 4.0

global H_IT_METH H_NI H_AS H_AL H_GW H_BT H_BTR H_SOT H_WSA ...
   H_STATUS H_STATUS1 H_STATUS2 ...
   H_GO H_BK H_RES H_RUPT H_MEN ...
   H_HIGE H_IT_BOX H_ASPECT H_ASPECT_EDIT ...
   S_PERF_INPUT S_USER_INPUT S_PERF_OUTPUT PICK REGIME

if nargin,
   switch Action
   case 'h_ni'
      set(H_NI,'Value',1)
      set(H_AS,'Value',0)
      set(H_AL,'Value',0)
      set(H_GW,'Value',0)
      set(H_BT,'Value',0)
      set(H_BTR,'Value',0)
      set(H_SOT,'Value',0)
      set(H_WSA,'Value',0)
      PICK=0;
   case 'h_as'
      set(H_NI,'Value',0)
      set(H_AS,'Value',1)
      set(H_AL,'Value',0)
      set(H_GW,'Value',0)
      set(H_BT,'Value',0)
      set(H_BTR,'Value',0)
      set(H_SOT,'Value',0)
      set(H_WSA,'Value',0)
      PICK=1;
   case 'h_al'
      set(H_NI,'Value',0)
      set(H_AS,'Value',0)
      set(H_AL,'Value',1)
      set(H_GW,'Value',0)
      set(H_BT,'Value',0)
      set(H_BTR,'Value',0)
      set(H_SOT,'Value',0)
      set(H_WSA,'Value',0)
      PICK=2;
   case 'h_gw'
      set(H_NI,'Value',0)
      set(H_AS,'Value',0)
      set(H_AL,'Value',0)
```

```
    set(H_GW,'Value',1)
    set(H_BT,'Value',0)
    set(H_BTR,'Value',0)
    set(H_SOT,'Value',0)
    set(H_WSA,'Value',0)
    PICK=3;
case 'h_bt'
    set(H_NI,'Value',0)
    set(H_AS,'Value',0)
    set(H_AL,'Value',0)
    set(H_GW,'Value',0)
    set(H_BT,'Value',1)
    set(H_BTR,'Value',0)
    set(H_SOT,'Value',0)
    set(H_WSA,'Value',0)
    PICK=4;
case 'h_btr'
    set(H_NI,'Value',0)
    set(H_AS,'Value',0)
    set(H_AL,'Value',0)
    set(H_GW,'Value',0)
    set(H_BT,'Value',0)
    set(H_BTR,'Value',1)
    set(H_SOT,'Value',0)
    set(H_WSA,'Value',0)
    PICK=5;
case 'h_sot'
    set(H_NI,'Value',0)
    set(H_AS,'Value',0)
    set(H_AL,'Value',0)
    set(H_GW,'Value',0)
    set(H_BT,'Value',0)
    set(H_BTR,'Value',0)
    set(H_SOT,'Value',1)
    set(H_WSA,'Value',0)
    PICK=6;
case 'h_wsa'
    set(H_NI,'Value',0)
    set(H_AS,'Value',0)
    set(H_AL,'Value',0)
    set(H_GW,'Value',0)
    set(H_BT,'Value',0)
    set(H_BTR,'Value',0)
    set(H_SOT,'Value',0)
    set(H_WSA,'Value',1)
    PICK=7;
case 'back'
    close (H_IT_METH)
    performance_input
case 'anal'
    set(H_GO,'Enable','off');
    set(H_RUPT,'Enable','on');
    set(H_BK,'Enable','off');
    set(H_RES,'Enable','off');
```

```
      set(H_MEN,'Enable','off');
      if get(H_NI,'Value')==1
         Perf
      elseif get(H_AS,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','AIRSPEED')
      elseif get(H_AL,'Value')==1
         iteration_parameters
         set(H_HIGE,'Enable','on')
         set(H_IT_BOX,'String','ALTITUDE')
      elseif get(H_GW,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','GROSS WEIGHT')
      elseif get(H_BT,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','BLADE TWIST')
      elseif get(H_BTR,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','BLADE TAPER RATIO')
      elseif get(H_SOT,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','START OF TAPER')
      elseif get(H_WSA,'Value')==1
         iteration_parameters
         set(H_IT_BOX,'String','WING SPAN AREA')
         set(H_ASPECT,'Enable','on')
         set(H_ASPECT_EDIT,'Enable','on')
      end
   case 'interrupt'
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','off');
      set(H_BK,'Enable','off');
      set(H_RES,'Enable','on');
      set(H_MEN,'Enable','on');
      uiwait;
   case 'resume'
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','on');
      set(H_BK,'Enable','off');
      set(H_RES,'Enable','off');
      set(H_MEN,'Enable','off');
      uiresume;
   case 'quit'
      quit_gui
   case 'return'
      close (H_IT_METH)
      janrad98
   case 'delta_input'
      close (H_IT_METH)
      performance_input
   case 'about'
      about_janrad
   end
end
```

APPENDIX N. ITERATION_PARAMETERS.M

This file creates GUI to enter iteration parameters. It is called by the Switchyard Callback function iteration_method_fcn.m.

```
function iteration_parameters()

%  GUI window to enter iterative steps.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load iteration_parameters

global H_IP H_HIGE H_IT_BOX H_ASPECT H_ASPECT_EDIT H_MEN

H_IP = figure('Units','normalized', ...
    'Color',[0.8 0.8 0.8], ...
    'Colormap',mat0, ...
    'Name','Iteration Parameters', ...
    'NumberTitle','off', ...
    'PoINTERShapeCData',mat1, ...
    'Position',[0.04375 0.0895833 0.875 0.78125], ...
    'Tag','Fig1');
b = uimenu('Parent',H_IP, ...
    'Label','JANRAD Options', ...
    'Tag','uimenu1');
c = uimenu('Parent',b, ...
    'Callback','iteration_parameters_fcn quit', ...
    'Label','Quit JANRAD', ...
    'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
    'Callback','iteration_parameters_fcn return', ...
    'Label','Return to Begining', ...
    'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
    'Callback','iteration_parameters_fcn delta_input', ...
    'Label','Change Input Parameters', ...
    'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
    'Callback','about_janrad', ...
```

```
    'Label','About Janrad 98 ...', ...
    'Separator','on', ...
    'Tag','Subuimenu1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.260714 0.888 0.476786 0.0533333], ...
    'String','Performance Analysis', ...
    'Style','text', ...
    'Tag','StaticText1');
H_IT_BOX = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.260714 0.824 0.476786 0.0533333], ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.2625 0.705778 0.357143 0.0533333], ...
    'String','Start Iteration at :', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.625 0.704 0.107143 0.0533333], ...
    'Style','edit', ...
    'Callback','global MINUM;MINUM=str2num(get(gcbo,''String''));',...
    'Tag','EditText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.2625 0.634667 0.357143 0.0533333], ...
    'String','End Iteration at :', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.625 0.634667 0.107143 0.0533333], ...
    'Style','edit', ...
    'Callback','global MAXUM;MAXUM=str2num(get(gcbo,''String''));',...
    'Tag','EditText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.2625 0.563556 0.357143 0.0533333], ...
    'String','Iteration Interval :', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
```

```
    'Position',[0.625 0.562667 0.107143 0.0533333], ...
    'Style','edit', ...
    'Callback','global INTER;INTER=str2num(get(gcbo,''String''));',...
    'Tag','EditText1');
H_ASPECT = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.2625 0.492444 0.355357 0.0533333], ...
    'String','Aspect Ratio :', ...
    'Style','text', ...
    'Enable','off',...
    'Tag','StaticText1');
H_ASPECT_EDIT = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.625 0.490667 0.107143 0.0533333], ...
    'Style','edit', ...
    'Enable','off',...
    'Callback','global AR;AR=str2num(get(gcbo,''String''));',...
    'Tag','EditText1');
H_HIGE = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Enable','off', ...
    'Position',[0.2625 0.421333 0.358929 0.0533333], ...
    'String','Include HIGE Calculations?', ...
    'Style','checkbox', ...
    'Value',0,...
    'Callback','if
get(gcbo,''Value'')==1,REGIME=1;else,REGIME=0;,end',...
    'Tag','Checkbox1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Callback','iteration_parameters_fcn back', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.260714 0.245333 0.196429 0.0986667], ...
    'String','<< BACK', ...
    'Tag','Pushbutton1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Callback','iteration_parameters_fcn anal', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.542857 0.245333 0.196429 0.0986667], ...
    'String','Analyze >>', ...
    'Tag','Pushbutton1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.180357 0.088 0.646429 0.106667], ...
    'String','Warning - Exessive Iteration Limits May Increase
Processing Times!', ...
```

```
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.176786 0.0746667 0.655357 0.125333], ...
    'Style','frame', ...
    'Tag','Frame1');
b = uicontrol('Parent',H_IP, ...
    'Units','normalized', ...
    'Position',[0.253571 0.810667 0.498214 0.146667], ...
    'Style','frame', ...
    'Tag','Frame2');
```

APPENDIX O. ITERATION_PARAMETERS_FCN.M

Switchyard Callback function for iteration_parameters.m GUI function.

```
function iteration_parameters_fcn(Action)

%  Switchyard Callback for iteration_parameters.m
%  JANRAD 98 VERSION 4.0

global H_IT_METH H_IP H_NI H_AS H_AL H_GW H_BT H_BTR H_SOT H_WSA H_HIGE
...
   H_GO H_BK H_RES H_RUPT H_MEN H_STATUS H_STATUS1 H_STATUS2...
   S_USER_INPUT PICK ...
   MINUM MAXUM INTER REGIME

if nargin,
   switch Action
   case 'back'
      set(H_BK,'Enable','on');
      set(H_GO,'Enable','on');
      set(H_RUPT,'Enable','off');
      set(H_RES,'Enable','off');
      set(H_MEN,'Enable','on');
      close(H_IP)
   case 'anal'
      set(H_BK,'Enable','off');
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','on');
      set(H_RES,'Enable','off');
      set(H_MEN,'Enable','off');
      close(H_IP)
      Perf
   case 'quit'
      quit_gui
   case 'return'
      janrad98
      close(H_IP)
      close (H_IT_METH)
   case 'delta_input'
      performance_input
      close (H_IP)
      close (H_IT_METH)
   case 'about'
      about_janrad
   end
end
```

APPENDIX P. OPTIONS.M

This file creates the GUI to select additional analysis methods and print input and output files saved from the performance output window.

```
function options()

%  GUI window to Select user options at end of performance routine.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load options

global H_OPTIONS H_PSCA H_PRDA H_CIM H_CID H_RTB H_EJANRAD NAME ...
   H_datain H_dataout H_vecdata ...
   H_printin H_printout H_printvec ...
   H_inputfile H_outputfile H_vecfile ...
   H_check1 H_check2 H_check3

H_OPTIONS = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'Name','Options', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[-0.003125 0.0625 0.954688 0.8625], ...
   'Tag','Fig1');
b = uimenu('Parent',H_OPTIONS,...
   'Label','JANRAD Options', ...
   'Tag','uimenu1');
c = uimenu('Parent',b, ...
   'Callback','options_fcn quit', ...
   'Label','Quit JANRAD', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','options_fcn return',...
   'Label','Return to Begining', ...
   'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
   'Callback','options_fcn delta_input',...
```

```
    'Label','Change Input Parameters', ...
    'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
    'Callback','options_fcn about',...
    'Label','About Janrad 98 ...', ...
    'Separator','on',...
    'Tag','Subuimenu1');
b = uicontrol('Parent',H_OPTIONS,...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',16, ...
    'FontWeight','bold', ...
    'Position',[0.0715631 0.808 0.378531 0.109333], ...
    'String','Select Option', ...
    'Style','text', ...
    'Tag','StaticText1');
H_PSCA = uicontrol('Parent',H_OPTIONS,...
    'Value',1,...
    'Callback','options_fcn h_psca',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0715631 0.72 0.378531 0.0533333], ...
    'String','Perform Stabilty & Control Analysis', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton3', ...
    'Value',1);
H_PRDA = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn h_prda',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0715631 0.632 0.378531 0.0533333], ...
    'String','Perform Rotor Dynamics Analysis', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton4');
H_CIM = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn h_cim',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0715631 0.541333 0.378531 0.0533333], ...
    'String','Change Iteration Method', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton1');
H_CID = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn h_cid',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0715631 0.453333 0.378531 0.0533333], ...
    'String','Change Input Data', ...
    'Style','radiobutton', ...
    'Tag','Radiobutton2');
H_RTB = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn h_rtb',...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
   'Position',[0.0715631 0.365333 0.378531 0.0533333], ...
   'String','Return to Begining', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton5');
H_EJANRAD = uicontrol('Parent',H_OPTIONS,...
   'Callback','options_fcn h_ejanrad',...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.0715631 0.274667 0.378531 0.0533333], ...
   'String','Exit JANRAD', ...
   'Style','radiobutton', ...
   'Tag','Radiobutton6');
b = uicontrol('Parent',H_OPTIONS,...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',16, ...
   'FontWeight','bold', ...
   'Position',[0.546139 0.805333 0.376648 0.106667], ...
   'String','Print Selection', ...
   'Style','text', ...
   'Tag','StaticText2');
H_printin = uicontrol('Parent',H_OPTIONS,...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.545548 0.71644 0.169908 0.0527903], ...
   'String','Print Input File :', ...
   'Style','checkbox', ...
   'Tag','Checkbox1', ...
   'Value',0);
H_inputfile = uicontrol('Parent',H_OPTIONS,...
   'Units','normalized', ...
   'FontSize',12, ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'String','',...
   'Position',[0.748209 0.71644 0.169908 0.0527903], ...
   'Style','text', ...
   'HorizontalAlignment','left',...
   'Tag','StaticText6');
H_printout = uicontrol('Parent',H_OPTIONS,...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Position',[0.545548 0.628959 0.169908 0.0527903], ...
   'String','Print Output File :', ...
   'Style','checkbox', ...
   'Tag','Checkbox1', ...
   'Value',0);
H_outputfile = uicontrol('Parent',H_OPTIONS,...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'String','',...
   'Position',[0.748209 0.628959 0.169908 0.0527903], ...
   'Style','text', ...
   'HorizontalAlignment','left',...
```

```
    'Tag','StaticText8');
H_printvec = uicontrol('Parent',H_OPTIONS,...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.545548 0.536953 0.169908 0.0527903], ...
    'String','Print Matrix & Vector File :', ...
    'Style','checkbox', ...
    'Tag','Checkbox1', ...
    'Value',0);
H_vecfile = uicontrol('Parent',H_OPTIONS,...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.748209 0.536953 0.169908 0.0527903], ...
    'String','',...
    'Style','text', ...
    'HorizontalAlignment','left',...
    'Tag','StaticText9');
b = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn print', ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.595104 0.402667 0.288136 0.072], ...
    'String','Send to Printer', ...
    'Tag','Pushbutton2');
b = uicontrol('Parent',H_OPTIONS,...
    'Units','normalized', ...
    'Position',[0.519774 0.362667 0.435028 0.570667], ...
    'Style','frame', ...
    'Tag','Frame1');
b = uicontrol('Parent',H_OPTIONS,...
    'Units','normalized', ...
    'Position',[0.0451977 0.0826667 0.440678 0.850667], ...
    'Style','frame', ...
    'Tag','Frame2');
b = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn back', ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.0809793 0.112 0.178908 0.088], ...
    'String','<< Back', ...
    'Tag','Pushbutton1');
b = uicontrol('Parent',H_OPTIONS,...
    'Callback','options_fcn cont', ...
    'Units','normalized', ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.286252 0.112 0.177024 0.088], ...
    'String','Continue >>', ...
    'Tag','Pushbutton1');

if get(H_check1,'Value')==0
```

```
   set(H_printin,'Enable','off')
end
if get(H_check2,'Value')==0
   set(H_printout,'Enable','off')
end
if get(H_check3,'Value')==0
   set(H_printvec,'Enable','off')
end

assignin('base','H_PSCA',H_PSCA);
assignin('base','H_PRDA',H_PRDA);
assignin('base','H_CIM',H_CIM);
assignin('base','H_CID',H_CID);
assignin('base','H_RTB',H_RTB);
assignin('base','H_EJANRAD',H_EJANRAD);
assignin('base','H_printin',H_printin);
assignin('base','H_printout',H_printout);
assignin('base','H_printvec',H_printvec);
assignin('base','H_inputfile',H_inputfile);
assignin('base','H_outputfile',H_outputfile);
assignin('base','H_vecfile',H_vecfile);
```

APPENDIX Q. OPTIONS_FCN.M

Switchyard Callback function for options.m GUI function.

```
function options_fcn(Action)

%  Switchyard Callback function for options.m
%  JANRAD 98 VERSION 4.0

global H_OPTIONS H_PSCA H_PRDA H_CIM H_CID H_RTB H_EJANRAD ...
   H_printin H_printout H_printvec...
   S_PERF_INPUT NAME S_MATR_VEC H_vecfile print_temp1


cond1=get(H_PSCA,'Value');
cond2=get(H_PRDA,'Value');
cond3=get(H_CIM,'Value');
cond4=get(H_CID,'Value');
cond5=get(H_RTB,'Value');
cond6=get(H_EJANRAD,'Value');

if nargin
   switch Action
   case 'h_psca'
      set(H_PSCA,'Value',1)
      set(H_PRDA,'Value',0)
      set(H_CIM,'Value',0)
      set(H_CID,'Value',0)
      set(H_RTB,'Value',0)
      set(H_EJANRAD,'Value',0)
   case 'h_prda'
      set(H_PSCA,'Value',0)
      set(H_PRDA,'Value',1)
      set(H_CIM,'Value',0)
      set(H_CID,'Value',0)
      set(H_RTB,'Value',0)
      set(H_EJANRAD,'Value',0)
   case 'h_cim'
      set(H_PSCA,'Value',0)
      set(H_PRDA,'Value',0)
      set(H_CIM,'Value',1)
      set(H_CID,'Value',0)
      set(H_RTB,'Value',0)
      set(H_EJANRAD,'Value',0)
   case 'h_cid'
      set(H_PSCA,'Value',0)
      set(H_PRDA,'Value',0)
      set(H_CIM,'Value',0)
      set(H_CID,'Value',1)
      set(H_RTB,'Value',0)
```

```
      set(H_EJANRAD,'Value',0)
   case 'h_rtb'
      set(H_PSCA,'Value',0)
      set(H_PRDA,'Value',0)
      set(H_CIM,'Value',0)
      set(H_CID,'Value',0)
      set(H_RTB,'Value',1)
      set(H_EJANRAD,'Value',0)
   case 'h_ejanrad'
      set(H_PSCA,'Value',0)
      set(H_PRDA,'Value',0)
      set(H_CIM,'Value',0)
      set(H_CID,'Value',0)
      set(H_RTB,'Value',0)
      set(H_EJANRAD,'Value',1)
   case 'back'
      close (H_OPTIONS)
      performance_output
   case 'print'
      if get(H_printin,'Value')==1,
         eval(['!copy ,print_temp, lpt1'])
         delete print_temp
      end
      if get(H_printout,'Value')==1,
         eval(['!copy ,print_temp1, lpt1'])
         delete print_temp1
      end
      if get(H_printvec,'Value')==1
         eval(['load ',get(H_vecfile,'String')]);
         diary print_temp2
         diary off
         delete print_temp2
         diary print_temp2
         r, psi, vi, theta, betat, alpha, Tpsi, Mpsi, DMpsi, dT, dM,
dD, cblade, CL, CD,
         diary off
         eval(['!copy /b ,print_temp2, lpt1'])
         delete print_temp2
      end
   case 'cont'
      if cond1==1
         stability_and_control
      elseif cond2==1
         rotor_dynamics
      elseif cond3==1
         close (H_OPTIONS)
         S_PERF_INPUT.Vinf=S_PERF_INPUT.Vinf/1.68894444;
         S_PERF_INPUT.twist=-S_PERF_INPUT.twist*57.3;
         S_PERF_INPUT.thetao=S_PERF_INPUT.thetao*57.3;
         iteration_method
      elseif cond4==1
         close (H_OPTIONS)
         performance_input
      elseif cond5==1
```

```
         close (H_OPTIONS)
         janrad98
      elseif cond6==1
         quit_gui
      else,
         error('Something is wrong in Options Switchyard Callback
Function')
      end
   case 'return'
      close all
      janrad98
   case 'delta_input'
      close (H_OPTIONS)
      performance_input
   case 'quit'
      quit_gui
   case 'about'
      about_janrad
   end
end
```

APPENDIX R. STABILITY_AND_CONTROL.M

This file creates figure window indicating the stability and control functions have not been incorporated.

```
function stability_and_control()

%  GUI Window to notify user that this module not installed.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load stability_and_control

a = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','Stability and Control Not Installed', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[0.190625 0.383333 0.446875 0.34375], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','close(gcf)', ...
   'FontSize',14, ...
   'FontWeight','bold', ...
   'Position',[0.388112 0.109091 0.202797 0.181818], ...
   'String','OK', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.0839161 0.515152 0.811189 0.345455], ...
   'String','The Stability and Control Function is not yet Avalilable
in JANRAD98', ...
   'Style','text', ...
   'Tag','StaticText1');
```

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.332168 0.357576 0.318182 0.127273], ...
    'String','SORRY!', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.013986 0.0363636 0.972028 0.933333], ...
    'Style','frame', ...
    'Tag','Frame1');
```

APPENDIX S. ROTOR_DYNAMICS.M

This file creates figure window indicating the rotor dynamics functions have not been incorporated.

```
function rotor_dynamics()

%  GUI Window to notify user that this module not installed.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load rotor_dynamics

a = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','Rotor Dynamics Not Installed', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[0.190625 0.3875 0.45625 0.339583], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','close(gcf)', ...
   'FontSize',14, ...
   'FontWeight','bold', ...
   'Position',[0.389078 0.153374 0.204778 0.184049], ...
   'String','OK', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.0821918 0.595092 0.849315 0.319018], ...
   'String','The Rotor Dynamics Function is not yet Avalilable in
JANRAD98', ...
   'Style','text', ...
   'Tag','StaticText1');
```

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.334471 0.429448 0.317406 0.122699], ...
    'String','SORRY!', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0205479 0.0306748 0.962329 0.93865], ...
    'Style','frame', ...
    'Tag','Frame1');
```

APPENDIX T. QUIT_QUI.M

This file creates GUI to verify the users intention to quit JANRAD 98.

```
function quit_gui()

%  GUI window to verify the users intention to quit Janrad98.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load quit_gui

a = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','''Quit JANRAD ''98?''', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[0.235938 0.433333 0.389063 0.266667], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','close (gcf)', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.188755 0.273438 0.240964 0.15625], ...
   'String','NO', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','close all,clear', ...
   'FontSize',12, ...
   'FontWeight','bold', ...
   'Position',[0.566265 0.273438 0.240964 0.15625], ...
   'String','YES', ...
   'Tag','Pushbutton2');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',12, ...
```

```
    'FontWeight','bold', ...
    'Position',[0.192771 0.59375 0.618474 0.289062], ...
    'String','Do You Really Want to Quit JANRAD 98?', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0401606 0.0625 0.907631 0.898438], ...
    'Style','frame', ...
    'Tag','Frame1');
```

APPENDIX U. TRIM_WARNING.M

This file creates GUI to inform user that the performance routine did not calculate a valid solution.

```
function trim_warning()

%  GUI window to notify user that conditions will not trim.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load trim_warning

a = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','WARNING!', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[0.184375 0.352083 0.5375 0.404167], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',14, ...
   'Position',[0.0755814 0.551546 0.825581 0.103093], ...
   'String','This configuration will not trim !', ...
   'Style','text', ...
   'Tag','StaticText1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','trim_warning_fcn', ...
   'FontSize',14, ...
   'FontWeight','bold', ...
   'Position',[0.373547 0.0927835 0.229651 0.164948], ...
   'String','OK', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
```

```
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0755814 0.340206 0.825581 0.170103], ...
    'String',mat2, ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'Position',[0.0755814 0.695876 0.825581 0.164948], ...
    'String','Performance Analysis Routine Terminated!', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0348837 0.0463918 0.924419 0.886598], ...
    'Style','frame', ...
    'Tag','Frame1');
```

APPENDIX V. TRIM_WARNING_FCN.M

Switchyard Callback for trim_warning.m GUI function.

```
function trim_warning_fcn()

%  Switchyard Callback function for trim_warning.m
%  JANRAD 98 VERSION 4.0

global H_GO H_RUPT H_BK H_RES H_MEN

set(H_GO,'Enable','off');
set(H_RUPT,'Enable','off');
set(H_BK,'Enable','on');
set(H_RES,'Enable','off');
set(H_MEN,'Enable','on');
close(gcf)
```

APPENDIX W. EMPTY_BOXES.M

This file creates GUI to inform user that all input edit boxes must contain a entry to properly execute performance evaluation.

```
function empty_boxes()

%  GUI window called if Empty edit boxes exist when continue button is
pressed.
%  JANRAD 98 VERSION 4.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

load empty_boxes

a = figure('Units','normalized', ...
   'Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'MenuBar','none', ...
   'Name','ERROR', ...
   'NumberTitle','off', ...
   'PointerShapeCData',mat1, ...
   'Position',[0.204688 0.35 0.451563 0.2875], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'Callback','close (gcf)', ...
   'FontSize',14, ...
   'FontWeight','bold', ...
   'Position',[0.401384 0.0942029 0.207612 0.202899], ...
   'String','OK', ...
   'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
   'Units','normalized', ...
   'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'FontSize',16, ...
   'FontWeight','bold', ...
   'Position',[0.107266 0.652174 0.795848 0.217391], ...
   'String','ERROR !', ...
   'Style','text', ...
   'Tag','StaticText1');
```

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',12, ...
    'Position',[0.107266 0.333333 0.795848 0.311594], ...
    'String','All edit boxes must contain a value to create a new
file.', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.0380623 0.0507246 0.930796 0.905797], ...
    'Style','frame', ...
    'Tag','Frame1');
```

APPENDIX X. JANRAD98_PLOTS.M

This script M-file creates plots for performance airspeed iteration evaluation. It is executed in Perf.m.

```
%  Script file called from Perf.m to plot airspeed iteration
%  performance results.
%  JANRAD 98 VERSION 4.0

%%%%    OUTPUT - change figure nbrs as needed   %%%%
%%%%    plotted only if airspeed iteration done %%%%
%%%%    add/comment out figures as needed       %%%%

if PICK==1

%%%% Main Rotor plots - Speed vs T/RHP/TPPangle/Liftpercent - 4 on 1
page %%%%
figure(11)
title('Lift to Drag Ratio for Entire Aircraft')
%subplot 221;plot(speed,LoverD); ylabel('W/De'); xlabel('Airspeed
(kts)')
subplot 221; plot(speed,thrust); ylabel('Thrust')
subplot 222; plot(speed,RHP); ylabel('Rotor Horsepower')
subplot 223; plot(speed,angle); ylabel('Tip Path Plane Angle (deg)')
subplot 224; plot(speed,Lperc); ylabel('Wing Lift Percentage')
%subplot 223; plot(mu,ctonsig)

%%%% Exact numbers for above graphs
disp('  Speed       W/De       RHP      TPP Angle     Lift Percent')
final=[speed LoverD RHP angle Lperc];

%%%% Total Pwr reqd/avail plot %%%%
%% first compute pwr avail - 0.91 accts for 9 percent losses fm eng pwr
output %%
%%  THIS CONTAINS ENGINE SPECIFIC PWR AVAIL NUMBERS  %%

SHP0 = 0.91*[6252; 5436; 4300];     % SLS     \
SHP0 = 0.91*[4037; 3511; 2777];  % SL Trop(108)  > use approptiate one
SHP0 = [4226; 3674; 2907];       % 8000 ft ISA  /   depending on
conditions
gamma = 1.4;
M = speed.*(6080.2/3600)./sqrt(gamma*1714*(temp+459.6)); % Mach nbr
delta = (1+(gamma-1)/2*M.^2).^((gamma-1)/gamma);
SHPav = SHP0*delta;     % Single Eng pwr avail
dSHPav = 2*SHPav;    % Dual Eng pwr avail
Sxmsn =  5040*ones(1,m);   % Single Eng Xmsn limit
Dxmsn =   7000*ones(1,m);  % Dual Eng Xmsn limit

figure(12)
```

```
plot(speed, Totpwr);%hold on;
plot(speed,SHPav(1,:),'--')
plot(speed,dSHPav(2,:),speed,dSHPav(3,:),'-.')
plot(speed,Dxmsn)
title('Total Power Required/Available')
xlabel('Airspeed (kts)')
ylabel('Shaft Horsepower')
axis([0 200 0 10000])
axis('square')


%%%% Fan-in-Tail Plots - Thrust vs Speed & Pwr vs Speed - on 1 page
%%%%
figure(13)
subplot 211; plot(speed, Tfan,speed,Tfin,'--')
title('Anti-Torque Thrust Required and Vertical Fin Thrust Provided')
ylabel('Thrust (lbs)')
subplot 212; plot(speed,pwrfani,'--',speed,pwrfanp,'-.',speed,pwrfant)
title('Fan Power Required')
xlabel('Airspeed (kts)')
ylabel('Power (hp)')

%%%% Fan-in-Tail Plot - Speed vs Tot Fan Pwr %%%%
figure(14)
plot(speed, pwrfant./RHP*100)
title('Tail Rotor Power Required as Percent of Rotor Power')
xlabel('Airspeed')
ylabel('Percent Rotor Power')

%%%% Plot of Coeff of Pwr vs Coeff of Thrust at Max Airspeed %%%%
%figure(15)
%plot(Tcoeff,Pcoeff)
%title('CP versus CT for Maximum Airspeed')
%xlabel('Thrust Coefficient (CT)')
%ylabel('Engine Power Coefficient (CP)')
%gtext('SLS Conditions')

%%%% Plot of Coefficient of Thrust vs Figure of Merit %%%%
%figure(16)
%plot(Tcoeff,figmrt)
%title('HOGE FM versus CT for SLS')
%xlabel('Thrust Coefficient (CT)')
%ylabel('Figure of Merit (FM)')

end
```

APPENDIX Y. STRUCTURE.M

This script M-file creates the input structure S_USER_INPUT.

```
%  Structure Consruction for JANRAD98 Performance_input.m
%  JANRAD 98 VERSION 4.0

S_USER_INPUT=struct(...
   'PA',PA,...                  % Pressure Altitude
   'temp',temp,...              % Air Temperature
   'Vinf',Vinf,...              % Airspeed
   'GW',GW,...                  % Gross Weight
   'omega',omega,...            % Rotational Velocity
   'naz',naz,...                % Number of Azimuths
   'thetao',thetao,...          % Collective Setting (deg)
   'Swing',Swing,...            % Wing Area
   'bwing',bwing,...            % Wing Span
   'CLwing',CLwing,...          % Wing CL
   'CDowing',CDowing,...        % Wing CDo
   'ewing',ewing,...            % Wing Efficiency Factor
   'afoil',afoil,...            % Airfoil Type
   'a',a,...                    % Blade Lift Curve Slope
   'b',b,...                    % Number of Rotor Blades
   'R',R,...                    % Rotor Radius
   'e',e,...                    % Effective Hinge Offset
   'grip',grip,...              % Length of Inner Non-Aerodynamic Portion
of Blade
   'rchord',rchord,...          % Root Chord
   'tr',tr,...                  % Rotor Blade Taper Ratio
   'trst',trst,...              % Rotor Blade Taper Ratio Starting Point
(r/R)
   'twist',twist,...            % Blade Twist
   'wblade',wblade,...          % Blade Weight
   'nbe',nbe,...                % Number of Blade Elements
   'Taux',Taux,...              % Aux Thrust
   'Afh',Afh,...                % Fuselage Equivalent Horizontal Flat
Plate Drag Area
   'Afv',Afv,...                % Vertical Projected Area (fuselage area
under disk)
   'Svert',Svert,...            % Area, Vertical Tail
   'bvert',bvert,...            % Span, Vertical Tail
   'CLvert',CLvert,...          % Lift Coefficient, Vertical Tail
   'CDovert',CDovert,...        % Profile Drag Coefficient, Vertical Tail
   'Shoriz',Shoriz,...          % Area, Horizontal Tail
   'bhoriz',bhoriz,...          % Span, Horizontal Tail
   'CLhoriz',CLhoriz,...        % Lift Coefficient, Horizontal Tail
   'CDohoriz',CDohoriz,...      % Profile Drag Coefficient, Horizontal
Tail
   'taildisk',taildisk);        % Tail Under Main Rotor (1-Yes, 2-No)
```

APPENDIX Z. STRUCTURE1.M

This script M-file creates the output structure S_PERF_OUTPUT.

```
%  Structure Consruction for JANRAD98 Performance_output.m
%  JANRAD 98 VERSION 4.0

S_PERF_OUTPUT=struct(...
   'Dfuse',Dfuse,...            %Fuselage drag
   'Hrotor',Hrotor,...          %Rotor drag
   'Lwing',Lwing,...            %Wing lift
   'Dwing',Dwing,...            %Wing drag
   'Lhoriz',Lhoriz,...          %Horizontal tail lifT
   'Dhoriz',Dhoriz,...          %Horizontal tail drag
   'Lvert',Lvert,...            %Vertical tail side force
   'Dvert',Dvert,...            %Vertical tail drag
   'alphaT',alphaT*57.3,...     %Tip path angle
   'betao',betao*57.3,...       %Rotor coning angle
   'rT2',rT2,...                %Location of mean thrust
   'thetao',thetao*57.3,...     %Rotor Collective pitch at .7 r/R
   'theta1c',theta1c*57.3,...   %1st lat cyclic term-A1 (deg)
   'theta1s',theta1s*57.3,...   %1st long cyclic term-B1 (deg)
   'solidity',solidity,...      %solidity (sigma)
   'DL',DL,...                  %Disk loading
   'FM',FM,...                  %Figure of Merit
   'CT_sig',CT_sig,...          %CT/sigma
   'CQ_sig',CQ_sig,...          %CQ/sigma
   'CH_sig',CH_sig,...          %CH/sigma
   'Machtip',Machtip,...        %Tip mach of the adv. blade
   'mu',mu,...                  %Advance ratio
   'T',T,...                    %Rotor thrust required (TPP)
   'Protor',Protor,...          %Rotor power required
   'Qrotor',Qrotor);            %Rotor torque
```

APPENDIX AA. STRUCTURE2.M

This script M-file creates the vector structure S_MATR_VEC.

```
%  Structure Consruction for JANRAD98 Perf.m
%  JANRAD 98 VERSION 4.0

S_MATR_VEC=struct(...
   'r',r,...                  %radial distance
   'psi',psi,...              %azimuth
   'vi',vi,...                %induced velocity
   'theta',theta,...          %collective pitch
   'betat',betat,...          %coning angle
   'alpha',alpha,...          %angle of attack
   'Tpsi',Tpsi,...            %Thrust at azimuth position
   'Mpsi',Mpsi,...            %Thrust Moment at azimuth position
   'DMpsi',DMpsi,...          %Drag Moment at azimuth position
   'dT',dT,...                %incremental Thrust
   'dM',dM,...                %incremental Thrust moment
   'dD',dD,...                %incremental drag
   'cblade',cblade,...        %blade chord
   'CL',CL,...                %incremental CL values
   'CD',CD);                  %incremental CD values
```

APPENDIX AB. UNSTRUCTURE.M

This script M-file decomposes the S_PERF_INPUT structure into 36 separate input parameter variables. These variables are corrected to proper units used in Perf.m computations.

```
%  Structure De-Consruction for JANRAD98 Perf.m
%  JANRAD 98 VERSION 4.0

   PA=S_PERF_INPUT.PA;
   temp=S_PERF_INPUT.temp;
   Vinf=S_PERF_INPUT.Vinf;
   GW=S_PERF_INPUT.GW;
   omega=S_PERF_INPUT.omega;
   naz=S_PERF_INPUT.naz;
   thetao=S_PERF_INPUT.thetao;
   Swing=S_PERF_INPUT.Swing;
   bwing=S_PERF_INPUT.bwing;
   CLwing=S_PERF_INPUT.CLwing;
   CDowing=S_PERF_INPUT.CDowing;
   ewing=S_PERF_INPUT.ewing;
   afoil=S_PERF_INPUT.afoil;
   a=S_PERF_INPUT.a;
   b=S_PERF_INPUT.b;
   R=S_PERF_INPUT.R;
   e=S_PERF_INPUT.e;
   grip=S_PERF_INPUT.grip;
   rchord=S_PERF_INPUT.rchord;
   tr=S_PERF_INPUT.tr;
   trst=S_PERF_INPUT.trst;
   twist=S_PERF_INPUT.twist;
   wblade=S_PERF_INPUT.wblade;
   nbe=S_PERF_INPUT.nbe;
   Taux=S_PERF_INPUT.Taux;
   Afh=S_PERF_INPUT.Afh;
   Afv=S_PERF_INPUT.Afv;
   Svert=S_PERF_INPUT.Svert;
   bvert=S_PERF_INPUT.bvert;
   CLvert=S_PERF_INPUT.CLvert;
   CDovert=S_PERF_INPUT.CDovert;
   Shoriz=S_PERF_INPUT.Shoriz;
   bhoriz=S_PERF_INPUT.bhoriz;
   CLhoriz=S_PERF_INPUT.CLhoriz;
   CDohoriz=S_PERF_INPUT.CDohoriz;
   taildisk=S_PERF_INPUT.taildisk;
```

APPENDIX AC. UNSTRUCTURE1.M

This script M-file decomposes the S_USER_INPUT structure into 36 separate input parameter variables. These variables are the actual values the user types or loads from a previously saved file

```
%  Structure De-Consruction for JANRAD98 Perf.m
%  JANRAD 98 VERSION 4.0

   PA=S_USER_INPUT.PA;
   temp=S_USER_INPUT.temp;
   Vinf=S_USER_INPUT.Vinf;
   GW=S_USER_INPUT.GW;
   omega=S_USER_INPUT.omega;
   naz=S_USER_INPUT.naz;
   thetao=S_USER_INPUT.thetao;
   Swing=S_USER_INPUT.Swing;
   bwing=S_USER_INPUT.bwing;
   CLwing=S_USER_INPUT.CLwing;
   CDowing=S_USER_INPUT.CDowing;
   ewing=S_USER_INPUT.ewing;
   afoil=S_USER_INPUT.afoil;
   a=S_USER_INPUT.a;
   b=S_USER_INPUT.b;
   R=S_USER_INPUT.R;
   e=S_USER_INPUT.e;
   grip=S_USER_INPUT.grip;
   rchord=S_USER_INPUT.rchord;
   tr=S_USER_INPUT.tr;
   trst=S_USER_INPUT.trst;
   twist=S_USER_INPUT.twist;
   wblade=S_USER_INPUT.wblade;
   nbe=S_USER_INPUT.nbe;
   Taux=S_USER_INPUT.Taux;
   Afh=S_USER_INPUT.Afh;
   Afv=S_USER_INPUT.Afv;
   Svert=S_USER_INPUT.Svert;
   bvert=S_USER_INPUT.bvert;
   CLvert=S_USER_INPUT.CLvert;
   CDovert=S_USER_INPUT.CDovert;
   Shoriz=S_USER_INPUT.Shoriz;
   bhoriz=S_USER_INPUT.bhoriz;
   CLhoriz=S_USER_INPUT.CLhoriz;
   CDohoriz=S_USER_INPUT.CDohoriz;
   taildisk=S_USER_INPUT.taildisk;
```

APPENDIX AD. UNSTRUCTURE2.M

This script M-file decomposes the S_PERF_OUTPUT structure into 25 separate output parameters. These variables are displayed in the performance output figure window.

```
% Structure De-Consruction for JANRAD98 performance_output.m
%  JANRAD 98 VERSION 4.0

   Dfuse=S_PERF_OUTPUT.Dfuse;
   Hrotor=S_PERF_OUTPUT.Hrotor;
   Lwing=S_PERF_OUTPUT.Lwing;
   Dwing=S_PERF_OUTPUT.Dwing;
   Lhoriz=S_PERF_OUTPUT.Lhoriz;
   Dhoriz=S_PERF_OUTPUT.Lhoriz;
   Lvert=S_PERF_OUTPUT.Lvert;
   Dvert=S_PERF_OUTPUT.Dvert;
   alphaT=S_PERF_OUTPUT.alphaT;
   betao=S_PERF_OUTPUT.betao;
   rT2=S_PERF_OUTPUT.rT2;
   thetao=S_PERF_OUTPUT.thetao;
   theta1c=S_PERF_OUTPUT.theta1c;
   theta1s=S_PERF_OUTPUT.theta1s;
   solidity=S_PERF_OUTPUT.solidity;
   DL=S_PERF_OUTPUT.DL;
   FM=S_PERF_OUTPUT.FM;
   CT_sig=S_PERF_OUTPUT.CT_sig;
   CQ_sig=S_PERF_OUTPUT.CQ_sig;
   CH_sig=S_PERF_OUTPUT.CH_sig;
   Machtip=S_PERF_OUTPUT.Machtip;
   mu=S_PERF_OUTPUT.mu;
   T=S_PERF_OUTPUT.T;
   Protor=S_PERF_OUTPUT.Protor;
   Qrotor=S_PERF_OUTPUT.Qrotor;
```

APPENDIX AE. UNSTRUCTURE3.M

This script M-file decomposes the S_MATR_VEC structure for use in the print and save commands.

```
%  Structure De-Consruction for JANRAD98 Perf.m
%  JANRAD 98 VERSION 4.0

r= S_MATR_VEC.r;
psi= S_MATR_VEC.psi;
vi= S_MATR_VEC.vi;
theta= S_MATR_VEC.theta;
betat= S_MATR_VEC.betat;
alpha= S_MATR_VEC.alpha;
Tpsi= S_MATR_VEC.Tpsi;
Mpsi= S_MATR_VEC.Mpsi;
DMpsi= S_MATR_VEC.DMpsi;
dT= S_MATR_VEC.dT;
dM= S_MATR_VEC.dM;
dD= S_MATR_VEC.dD;
cblade= S_MATR_VEC.cblade;
CL= S_MATR_VEC.CL;
CD= S_MATR_VEC.CD;
```

APPENDIX AF. PERF.M

This script M-file is the primary computational routine for JANRAD 98. It is launched from either the iteration_method_fcn.m or iteration_parameter_fcn.m Switchyard Callback function.

```
%  Perf.m
%  Main Performance computaion routine.
%  JANRAD 98 VERSION 4.0

global S_PERF_INPUT S_MATR_VEC

if S_PERF_INPUT.grip < 1e-10,
   S_PERF_INPUT.grip=1e-10;
end

if S_PERF_INPUT.Swing < 1e-10,
   S_PERF_INPUT.Swing=1e-10;
end

if S_PERF_INPUT.bwing < 1e-10,
   S_PERF_INPUT.bwing=1e-10;
end

if S_PERF_INPUT.ewing < 1e-10,
   S_PERF_INPUT.ewing=1e-10;
end

if S_PERF_INPUT.Shoriz < 1e-10,
   S_PERF_INPUT.horiz=1e-10;
end

if S_PERF_INPUT.bhoriz < 1e-10,
   S_PERF_INPUT.bhoriz=1e-10;
end

if S_PERF_INPUT.Svert < 1e-10,
   S_PERF_INPUT.Svert=1e-10;
end

if S_PERF_INPUT.bvert < 1e-10,
   S_PERF_INPUT.bvert=1e-10;
end

S_PERF_INPUT.Vinf=S_PERF_INPUT.Vinf*1.68894444;
S_PERF_INPUT.twist=abs(S_PERF_INPUT.twist)/57.3;
S_PERF_INPUT.thetao=S_PERF_INPUT.thetao/57.3;
```

```
unstructure

fid=fopen('print_temp','w+');
fprintf(fid,'\t                    *** MODIFIED USER INPUT ***\n\n');
fprintf(fid,'\t                      Forward velocity = %6.0f
kts\n',Vinf/1.69);
fprintf(fid,'\t                           Temperature = %6.0f degs
F\n',temp);
fprintf(fid,'\t                     Pressure altitude = %6.0f ft\n',PA);
fprintf(fid,'\t                          Gross weight = %6.0f lbs\n',GW);
fprintf(fid,'\t                     Number of blades = %6.0f \n',b);
fprintf(fid,'\t                          Rotor radius = %6.2f ft\n',R);
fprintf(fid,'\t                     Blade root chord = %6.2f
ft\n',rchord);
fprintf(fid,'\t                           Blade twist = %6.2f degs\n',-
1*twist*57.3);
fprintf(fid,'\t                Blade lift curve slope = %6.2f \n',a);
fprintf(fid,'\t                           Blade weight = %6.2f
lbs\n',wblade);
fprintf(fid,'\t                   Rotational velocity = %6.2f
rads/sec\n',omega);
fprintf(fid,'\t                     Blade grip length = %6.2f ft\n',grip);
fprintf(fid,'\t                          Hinge offset = %6.2f ft\n',e);
fprintf(fid,'\t            Equivalent flat plate area = %6.2f
ft^2\n',Afh);
fprintf(fid,'\t               Vertical projected area = %6.2f
ft^2\n',Afv);
fprintf(fid,'\t                             Wing area = %6.2f
ft^2\n',Swing);
fprintf(fid,'\t                             Wing span = %6.2f
ft\n',bwing);
fprintf(fid,'\t                               Wing CL = %6.2f \n',CLwing);
fprintf(fid,'\t                              Wing CDo = %6.4f
\n',CDowing);
fprintf(fid,'\t                 Wing efficiency factor = %6.2f \n',ewing);
fprintf(fid,'\t                   Horizontal tail area = %6.2f
ft^2\n',Shoriz);
fprintf(fid,'\t                   Horizontal tail span = %6.2f
ft\n',bhoriz);
fprintf(fid,'\t                     Horizontal tail CL = %6.2f
\n',CLhoriz);
fprintf(fid,'\t                    Horizontal tail CDo = %6.4f
\n',CDohoriz);
fprintf(fid,'\t                     Vertical tail area = %6.2f
ft^2\n',Svert);
fprintf(fid,'\t                     Vertical tail span = %6.2f
ft\n',bvert);
fprintf(fid,'\t                       Vertical tail CL = %6.2f \n',CLvert);
fprintf(fid,'\t                      Vertical tail CDo = %6.4f
\n',CDovert);
fprintf(fid,'\t                       Auxiliary thrust = %6.0f
lbs\n',Taux);
fprintf(fid,'\t                     Number of Azimuths = %6.0f \n',naz);
```

```
fprintf(fid,'\t                    Collective Pitch = %6.2f
degs\n',thetao*57.3);
fprintf(fid,'\t                        Airfoil Type = %6.0f \n',afoil);
fprintf(fid,'\t                         Taper Ratio = %6.2f \n',tr);
fprintf(fid,'\t                Taper Ratio Starts At = %6.2f ft\n',trst);
fprintf(fid,'\t            Number of Blade Elements = %6.0f \n',nbe);
fprintf(fid,'\t Tail Under Main Rotor (1-yes 2-no) = %6.0f
\f',taildisk);
fclose(fid);

switch PICK
case 1
    if isempty(MINUM)
        MINUM=0;      % sets min airspeed default to 0 kts
    end
    if isempty(MAXUM)
        MAXUM=160;       % sets max airspeed default to 160 kts
    end
    if isempty(INTER)
        INTER=20;     % sets INTERval default to 20 kts
    end
    if MINUM > MAXUM % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
    doit1='airspd=itervar;';
    doit2='Vinf=airspd*6080.2/3600;';  % converts kts to ft/s
    m=(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    speed=zeros(1,m);       % vector for each airspeed values
    mu=zeros(1,m);       % vector for adv ratio values
    Lperc=zeros(1,m);       % vector for wing lift percent values
    ctonsig=zeros(1,m); % vector for CT/sigma values
    LoverD=zeros(1,m);      % vector for tot acft W/De
    theone='airspd';

case 2
    if isempty(MINUM)
        MINUM=0;      % sets min altitude default to 0 ft
    end

    if isempty(MAXUM)
        MAXUM=15000;     % sets max altitude default to 15000 ft
    end

    if isempty(INTER),
        INTER=1000;      % sets INTERval default to 1000 ft
    end
    if MINUM > MAXUM     % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
```

```
    doit1='PA=itervar;';
    % sets correct ISA temp
    % for given alt (deg F)
    % note: 59=SLS, 103=trop
    doit2='temp = 59-1.9811e-3/.5555*PA;';
    m=(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    if REGIME==1
        Ptige=zeros(1,m);
    end
    altitude=zeros(1,m);
    theone='PA';

case 3
    if isempty(MINUM)
        MINUM=10000;     % sets min GW default to 10000 lbs
    end
    if isempty(MAXUM)
        MAXUM=20000;     % sets max airspeed default to 20000 lbs
    end

    if isempty(INTER),
        INTER=2000;      % sets INTERval default to 2000 lbs
    end
    if MINUM > MAXUM     % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
    doit1='GW=itervar;';
    doit2=' ';
    m=(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    wt=zeros(1,m);       % vector for GW values
    Tcoeff=zeros(1,m);      % vector for Coeff of Thrust values
    Pcoeff=zeros(1,m);      % vector for Coeff of Power values
    figmrt=zeros(1,m);      % vector for Figure of Merit values
    theone='GW';

case 4
    if isempty(MINUM)
        MINUM=0;      % sets min blade twist default to 0 deg
    end
    if isempty(MAXUM)
        MAXUM=-12;       % sets max blade twist default to -12 deg
    end
    if isempty(INTER),
        INTER=-2;     % sets INTERval default to -2 deg
    end
    if MINUM > MAXUM % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
```

```
    doit1='TWIST=itervar;';
    doit2='twist=TWIST/57.3;';     % converts degrees to radians
    m=abs(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    thetat=zeros(1,m);       % vector for blade twist values
    theone='TWIST';

case 5
    if isempty(MINUM)
        MINUM=1;     % sets min blade taper ratio default to 1
    end
    if isempty(MAXUM)
        MAXUM=0.5;      % sets max blade taper ratio default to 0.5
    end
    if isempty(INTER),
        INTER=-0.1;     % sets INTERval default to -0.1
    end
    if MINUM > MAXUM    % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
    doit1='tr=itervar;';
    doit2=' ';
    m=abs(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    taper=zeros(1,m);      % vector for blade taper ratio values
    theone='tr';

case 6
    if isempty(MINUM)
        MINUM=0.1;      % sets min blade twist default to 0.1 r/R
    end
    if isempty(MAXUM)
        MAXUM=0.9;      % sets max blade twist default to 0.9 r/R
    end
    if isempty(INTER),
        INTER=0.1;      % sets INTERval default to 0.1 r/R
    end
    if MINUM > MAXUM % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
    doit1='trst=itervar;';
    doit2=' ';
    m=abs(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    start=zeros(1,m);      % vector for taper ratio start position
values
    theone='trst';

case 7
    if isempty(MINUM)
```

```
        MINUM=50;     % sets min wing area default to 50 sq ft
    end
    if isempty(MAXUM)
        MAXUM=80;     % sets max wing area default to 80 sq ft
    end
    if isempty(INTER),
        INTER=5;      % sets INTERval default to 5 sq ft
    end
    if MINUM > MAXUM    % ensures INTERval is the correct sign
        INTER=-abs(INTER);
    else
        INTER=abs(INTER);
    end
    if isempty(AR)
        AR=6;      % sets aspect ratio default to 6
    end
    doit1='Swing=itervar;';
    doit2='bwing=sqrt(AR*Swing);';  % computes wing span given AR and
wing area
    m=abs(MAXUM-MINUM)/INTER+1;  % computes reqd nbr of elements for
vectors
    area=zeros(1,m);        % vector for wing area values
    theone='Swing';

case 0
    MINUM=1;    %\
    MAXUM=1;    % > using these values the for loop is effectively
    INTER=1;    %/          non-existent - will go through only once
    doit1=' ';
    doit2=' ';
end

%%% common vectors for compilation of iterative data %%%

if PICK~=0
    thrust=zeros(1,m);
    RHP=zeros(1,m);
    Totpwr=zeros(1,m);
    angle=zeros(1,m);
    Drag=zeros(1,m);
end

%%% account for vertical drag on wing and horizontal tail %%%

Afv1=Afv;   % puts the vert profile area into temp variable Afv1

if taildisk==1
    Afv=Afv1+3*(Swing+Shoriz);   % this assumes a vertical Cd of 1.2
for the
elseif taildisk==2          % wing and horiz tail and a vertical Cd of
    Afv=Afv1+3*Swing;       % 0.4 for the fuselage (i.e. 1.2/.4 = 3)
end                         % thus making wing/tail effectively larger
                            % when hvr thrust calc using Cd=0.4
%%% NOTE:  All total power calculations are based on a Fan-in-Tail %%%
```

```
%%%         anti-torque device.  These equations must be modified  %%%
%%%         if using a conventional tail rotor or NOTAR system     %%%

%%% Fan-in-Tail Parameters %%%

Afan = pi*3.5^2;      % Area of fan; radius=3.5 ft
lfan = 41;         % Distance fm M/R hub to fan hub
sigmafan = 0.143;     % Solidity of fan
cdfan = 0.015;        % Cdo of fan blade

%%% vectors for Fan-in-Tail calculations %%%

if PICK~=0
    Tfin=zeros(1,m);
    Tfan=zeros(1,m);
    pwrfani=zeros(1,m);
    pwrfanp=zeros(1,m);
    pwrfant=zeros(1,m);
end

%%% Beginning of Iteration Loop %%%
p=0;
for itervar = MINUM:INTER:MAXUM
    eval(doit1)    % reads and evaluates the string 'doit1' assigned
above
    % assigns itervar to the reqd var needed to complete iteration
    eval(doit2)    % reads and evaluates the string 'doit2' assigned
above
    % used if input variable needs some type of manipulation
    p=p+1;      % used to move to next vector element
    set(H_STATUS,'String','ROTOR PERFORMANCE ROUTINE')
    tic
    set(H_STATUS1,'String','Start Elapsed Time')
    pause(3)
    trim

    %%%   *** Calculation of output parameters ***    %%%

    load perftemp    % Eccles addition - program was not recognizing
                     % mchord and DMpsi.
    Protor=mean(DMpsi)*b*omega/550; % Rotor pwr reqd, NOT total pwr
    Qrotor=mean(DMpsi)*b;      % Rotor torque reqd
    solidity=b*mchord/(pi*R);
    CQ=Qrotor/(Adisk*rho*Vtip^2*R);
    CH=Hrotor/(Adisk*rho*Vtip^2);
    CT_sig=CT/solidity;
    CQ_sig=CQ/solidity;
    CH_sig=CH/solidity;
    Machtip=(Vtip*cos(alphaT)+Vinf)/(49.05*sqrt(temp+460));
    if Vinf < 16.9,
        DL=T/(pi*R^2);
        FM=(T*sqrt(DL/(2*rho)))/(550*Protor);
    else
        DL=0;
```

```
        FM=0;
    end

    %%%% Compute Fan and Access/Xmsn Power Reqd in Fwd Flt %%%%
    vifan = abs(sqrt((Qrotor/lfan-Lvert)/(rho*Afan)));   % use abs in
case complex
    (Qrotor/lfan-Lvert<0);
    Thrustf = Qrotor/lfan;          % Thrust reqd for anti-torque
    Pfani = 0.5*rho*Afan*vifan^3/550;     % Fan induced pwr
    Pfanp = rho*Afan*Vtip^3*sigmafan*cdfan/4400;   % Fan profile pwr
    Pacc = 106.25+0.01275*Protor;         % Pwr for accessories
    Ptot = Protor+Pfani+Pfanp+Pacc;       % Total pwr reqd
    CP=Ptot*550/(Adisk*rho*Vtip^3);       % Coefficient of Pwr based on
Ptot

    %%%% HIGE total pwr %%%%
    if REGIME==1
        viave=sqrt(DL/(2*rho));         % Induced vel - hoge
        delpwr=T*viave*0.23/550;        % Decr in pwr due to grd effect
        Prtrige=Protor-delpwr;          % Rotor pwr - hige
        Tige=Prtrige*550/(omega*lfan);    % Rotor thrust - hige
        vifige=sqrt(Tige/(rho*Afan));       % Induce vel - hige
        Pfanige=0.5*rho*Afan*vifige^3/550;      % Fan pwr - hige
        Paccige=106.25+0.01275*Prtrige;     % Access pwr - hige
        Ptotige=Prtrige+Pfanige+Pfanp+Paccige;  % Tot pwr - hige
    end

    WonDe=GW/(550*Ptot/Vinf);    % Total acft W/De (lift/drag)

%%% Collecting and vectoring all the calculated data %%%
%%% vectors specific to desired iteration

switch PICK
case 1
    speed(p)=airspd; % vector of airspeeds
    mu(p)=Vinf/Vtip; % vector of advance ratios
    Lperc(p)=Lwing/GW*100;% vector wing lift percentage
    ctonsig(p)=CT_sig;  % vector of CT on sigma
    LoverD(p)=WonDe; % vector of W/De

case 2
    altitude(p)=PA;  % vector of altitudes

case 3
    wt(p)=GW;           % vector of GW
    Tcoeff(p)=CT;    % vector of Coefficient of Thrust
    Pcoeff(p)=CP;    % vector of Coefficient of Power
    figmrt(p)=FM;    % vector of Figure of Merit

case 4
    thetat(p)=TWIST; % vector of blade twist

case 5
    taper(p)=tr;     % vector of taper ratio
```

```
case 6
    start(p)=trst;    % vector of starting position of blade taper

case 7
    area(p)=Swing;    % vector of wing wetted area
end

if REGIME==1
    Ptige(p)=Ptotige;    % vector of total hp in hige
end

%%% vectors common to all iterations

if PICK~=0
    thrust(p)=T;      % vector of main rotor thrust
    RHP(p)=Protor;    % vector of rotor hp reqd
    Totpwr(p)=Ptot;   % vector of total hp reqd
    angle(p)=alphaT*57.3;  % vector of tip path plane angle
    Drag(p)=Drotor+Dfuse+Dvert+Dhoriz+Dwing; % vector of tot acft drag
    Tfin(p)=Lvert;    % vector of vertical fin lift
    Tfan(p)=Qrotor/lfan;    % vector of fan thrust
    pwrfani(p)=Pfani;    % vector of fan induced pwr
    pwrfanp(p)=Pfanp;    % vector of fan profile pwr
    pwrfant(p)=Pfani+Pfanp;    % vector of fan total pwr
    eval('theone');  % displays current value of iterative variable
end         %  'end' needed to complete the 'for' loop

end   % this is the 'end' needed to complete the 'for' loop

%%%% Save iteration data for future processing %%%%
%%% format: save <filename> var1 var2 var3 %%%
%%% (note: all variables must be valid or will get error) %%%
%%% Works well to create short m-file to graph this calculated data %%%
%%%   just use 'load <filename>' at the beginning of the file to %%%
%%%   read all the vectors which are stored in the .mat file %%%

if PICK~=0
    save output Totpwr angle RHP thrust Drag
    save tailop Tfin Tfan pwrfani pwrfanp pwrfant
end

switch PICK
case 1
    save extra1 speed mu ctonsig LoverD Lperc RHP

case 2
    save extra2 altitude

case 3
    save extra3 wt Tcoeff Pcoeff figmrt

case 4
    save extra4 thetat
```

```
case 5
    save extra5 taper

case 6
    save extra6 start

case 7
    save extra7 area
end

if REGIME==1
    save extra9 Ptige
end

if PICK==1
    janrad98_plots
end

set(H_STATUS,'String','COMPLETING CALCULATIONS ...')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

%   *** output to disk (text file) ***

fid=fopen('print_temp1','w+');
fprintf(fid,'\t                  *** RESULTS ***\n\n');
fprintf(fid,'\t              Forward velocity = %6.0f kts\n',Vinf/1.69);
fprintf(fid,'\t                   Temperature = %6.0f degs F\n',temp);
fprintf(fid,'\t             Pressure altitude = %6.0f ft\n',PA);
fprintf(fid,'\t                  Gross weight = %6.0f lbs\n',GW);
fprintf(fid,'\t              Number of blades = %6.0f \n',b);
fprintf(fid,'\t                  Rotor radius = %6.2f ft\n',R);
fprintf(fid,'\t              Blade mean chord = %6.2f ft\n',mchord);
fprintf(fid,'\t                   Blade twist = %6.2f degs\n',-
1*twist*57.3);
fprintf(fid,'\t       Blade lift curve slope = %6.2f \n',a);
fprintf(fid,'\t                  Blade weight = %6.2f lbs\n',wblade);
fprintf(fid,'\t          Rotational velocity = %6.2f
rads/sec\n',omega);
fprintf(fid,'\t            Blade grip length = %6.2f ft\n',grip);
fprintf(fid,'\t                 Hinge offset = %6.2f ft\n',e);
fprintf(fid,'\t   Equivalent flat plate area = %6.2f ft^2\n',Afh);
fprintf(fid,'\t      Vertical projected area = %6.2f ft^2\n',Afv);
fprintf(fid,'\t                   Wing area = %6.2f ft^2\n',Swing);
fprintf(fid,'\t                   Wing span = %6.2f ft\n',bwing);
fprintf(fid,'\t                     Wing CL = %6.2f \n',CLwing);
fprintf(fid,'\t                    Wing CDo = %6.4f \n',CDowing);
fprintf(fid,'\t       Wing efficiency factor = %6.2f \n',ewing);
fprintf(fid,'\t         Horizontal tail area = %6.2f ft^2\n',Shoriz);
fprintf(fid,'\t         Horizontal tail span = %6.2f ft\n',bhoriz);
fprintf(fid,'\t           Horizontal tail CL = %6.2f \n',CLhoriz);
fprintf(fid,'\t          Horizontal tail CDo = %6.4f \n',CDohoriz);
```

```
fprintf(fid,'\t           Vertical tail area = %6.2f ft^2\n',Svert);
fprintf(fid,'\t           Vertical tail span = %6.2f ft\n',bvert);
fprintf(fid,'\t             Vertical tail CL = %6.2f \n',CLvert);
fprintf(fid,'\t            Vertical tail CDo = %6.4f \n',CDovert);
fprintf(fid,'\t                Fuselage drag = %6.0f lbs\n',Dfuse);
fprintf(fid,'\t                   Rotor drag = %6.0f lbs\n',Hrotor);
fprintf(fid,'\t                    Wing lift = %6.0f lbs\n',Lwing);
fprintf(fid,'\t                    Wing drag = %6.0f lbs\n',Dwing);
fprintf(fid,'\t         Horizontal tail lift = %6.0f lbs\n',Lhoriz);
fprintf(fid,'\t         Horizontal tail drag = %6.0f lbs\n',Dhoriz);
fprintf(fid,'\t     Vertical tail side force = %6.0f lbs\n',Lvert);
fprintf(fid,'\t           Vertical tail drag = %6.0f lbs\n',Dvert);
fprintf(fid,'\t             Auxiliary thrust = %6.0f lbs\n',Taux);
fprintf(fid,'\t               Tip path angle = %6.2f
degs\n',alphaT*57.3);
fprintf(fid,'\t           Rotor coning angle = %6.2f
degs\n',betao*57.3);
fprintf(fid,'\tLocation of mean thrust (r/R) = %6.2f \n',rT2);
fprintf(fid,'\t   Collective pitch at .7 r/R = %6.2f
degs\n',thetao*57.3);
fprintf(fid,'\t 1st lat cyclic term-A1 (deg) = %6.2f \n',theta1c*57.3);
fprintf(fid,'\t1st long cyclic term-B1 (deg) = %6.2f \n',theta1s*57.3);
fprintf(fid,'\t                   solidity = %6.3f \n',solidity);
fprintf(fid,'\t                Disk loading = %6.2f lbs/ft^2\n',DL);
fprintf(fid,'\t              Figure of Merit = %6.2f \n',FM);
fprintf(fid,'\t                     CT/sigma = %6.3f \n',CT_sig);
fprintf(fid,'\t                     CQ/sigma = %6.4f \n',CQ_sig);
fprintf(fid,'\t                     CH/sigma = %6.4f \n',CH_sig);
fprintf(fid,'\t   Tip mach of the adv. blade = %6.3f \n',Machtip);
fprintf(fid,'\t                Advance ratio = %6.3f \n',mu);
fprintf(fid,'\t  Rotor thrust required (TPP) = %6.0f lbs\n',T);;
fprintf(fid,'\t        Rotor power required = %6.0f h.p.\n',Protor);
fprintf(fid,'\t                Rotor torque = %6.0f ft-lbs\f',Qrotor);
fclose(fid);

%   *** Configuring variables for output ***

theta=theta*57.3;
betat=[betat twist*(0.7-(Reff+(R-Reff)/2)/R)]*57.3;
alpha=alpham*57.3;,alpha=[alpha zeros(size(psi))];
Mpsi=Mpsi(:,length(Mpsi(1,:))-1);
dM=[dM ddM];
psi=psi*57.3;
r=[r (R-(R-Reff)/2)];
vi=[vi 0];

set(H_STATUS,'String','STAND BY FOR OUTPUT...')
pause(3)

structure1
structure2
set(H_STATUS,'String','')
performance_output
close(H_IT_METH)
```

APPENDIX AG. TRIM.M

This script M-file is a subroutine to trim the rotor system. It is called in Perf.m.

```
%  Trim.m

%  Trim routine for collective/cyclic.
%  JANRAD 98 VERSION 4.0

set(H_STATUS,'String','EXECUTING ROTOR TRIM ROUTINE')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

%    *** calculation of required parameters ***

rho=.002377*(-.000031*PA+(-.002*temp+1.118));

%    *** first guess at rotor profile drag ( H force) ***

if Vinf < 16.9
   Drotor=0;
else
   Drotor=Vinf*(rho/.002377);
end

q=0.5*rho*Vinf^2;
Adisk=pi*R^2;
Vtip=omega*R;
Dfuse=q*Afh;
CDwing=CDowing+(CLwing^2/(ewing*pi*(bwing^2/Swing)));
CDhoriz=CDohoriz+(CLhoriz^2/(.8*pi*(bhoriz^2/Shoriz)));
CDvert=CDovert+(CLvert^2/(.8*pi*(bvert^2/Svert)));
Dwing=q*CDwing*Swing;
Dhoriz=q*CDhoriz*Shoriz;
Dvert=q*CDvert*Svert;
Dftotal=(Dfuse+Dwing+Dhoriz+Dvert)-Taux;
Lwing=q*CLwing*Swing;
Lhoriz=q*CLhoriz*Shoriz;
Lvert=q*CLvert*Svert;
Lftotal=Lwing+Lhoriz;
alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
mu=Vinf*cos(alphaT)/Vtip;

%    *** thrust calculation ***

if Vinf < 16.9
   T=(1+(0.4*Afv/Adisk))*GW;
   CT=T/(Adisk*rho*Vtip^2);
else
```

```
   T=(GW-Lftotal)/cos(alphaT);
   T=T/(Adisk*rho*Vtip^2);
end

%   *** setup blade radius elements, azimuth elements,
%       induced velocity distributions, and determination
%       of coning angle and tip loss parameter ***

B=1-(sqrt(2*CT)/b);
Reff=B*R;
Rbar=Reff-e;
dr=(Reff-grip)/nbe;
r=grip:dr:Reff-dr;,r=r+dr/2;
rT1=0.7;% *** first guess at rT ***
RbarT=rT1*Rbar;
mblade=wblade/32.17;
betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade));
betat=twist*(0.7-(r/R));
psi=0:360/naz:360-360/naz;,psi=psi'/57.3;

%% set up vector of blade element chords and then varies them as
%%  requested with the blade taper and blade taper start position
%%  rchord=root chord
%%  cblade=vector of blade element chord lengths
%%  tr=taper ratio (tip/root)
%%  trst=taper ratio start position (r/R)

cblade=rchord*ones(size(r));  % gives all elements same chord length
initially

if tr==0 % prevents division by zero later in code
    tr=1;    %  in case 0 is enter for taper ratio instead
end      %  of 1 for no taper

if trst==0
    slope=(rchord-rchord*tr)/(Reff-grip); % Modifies each element
    cblade=cblade-slope*(r-grip);       %  chord length wrt input
    tchord=cblade(nbe);           %  taper ratio which has been
    mchord=sum(cblade)/nbe;          %  been converted into a slope
    %  top portion takes into
else                    %  account the possibility that
    slope=(rchord-rchord*tr)/(R*(1-trst));   %  a 0 start position is
really at
    z=fix(nbe*trst);           %  the start of the aero portion
    if z<=1    % prevents beginning index fm being zero
        z=1;
    end
    cblade(z:nbe)=cblade(z:nbe)-(r(z:nbe)-r(z))*slope;
    tchord=cblade(nbe);
    mchord=sum(cblade)/nbe;
end
```

```
%   *** induced velocity determination ***

if Vinf < 16.9,
    A=4*pi;
    Bv=(b/2)*omega*a.*cblade;
    Tv=0;
    delT=T-Tv;
    while abs(delT) > .01*T   % Prouty Eqns for Hover
        thetav=twist*(0.7-(r/R))+thetao;
        C=(-b/2).*cblade*omega^2.*r*a.*thetav;
        vi=(-Bv+sqrt(Bv.^2-(4*A*C)))/(2*A);
        dTv=(b/2)*rho*((omega*r).^2)*a.*(thetav-
(vi./(omega*r))).*cblade*dr;
        Tv=sum(dTv);
        delT=T-Tv;
        if delT < 0,
            thetao=thetao-0.5*thetao*abs(delT/T);
        else
            thetao=thetao+0.5*thetao*abs(delT/T);
        end
    end
else        % Wheatley Eqn for Fwd flt
    lamdaT=0;
    lamda=1;
    while abs(lamdaT-lamda)>1e-4
        lamda=lamdaT;
        lamdaT=mu*sin(alphaT)+0.5*CT/sqrt(lamdaT^2+mu^2);
    end
    vi=lamdaT*Vtip-Vinf*sin(alphaT);
    vi=vi*ones(size(r));
end

%   *** first guess at theta ***

theta1c=0.035*((0.0006e-3*Vinf^2+0.244e-3*Vinf)/0.105);
theta1s=-0.087*((0.0006e-3*Vinf^2+0.244e-3*Vinf)/0.105);
theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

%   *** rotor trimming routine ***

set(H_STATUS,'String','TRIMMING COLLECTIVE')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

k=1;
error0=(T*.02)+1;

while abs(error0) > T*.02
    Tpsi=zeros(size(psi));
    thrcalc

    if k>1,   % Eccles change: These three lines were added.
        error1;
```

```
    end

    error0=T-(mean(Tpsi)*b);
    if error0 < -T*.02,
        thetao=thetao-0.35*thetao*abs(1.5*error0/T)*(1-mu);
    elseif error0 > T*.02,
        thetao=thetao+0.35*thetao*abs(1.5*error0/T)*(1-mu);
    end
    theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);
    if k > 1,
        if abs(error0) > abs(error1),
            clc
            trim_warning
            set(H_GO,'Enable','off');
            set(H_RES,'Enable','off');
            set(H_RUPT,'Enable','off');
            set(H_BK,'Enable','on');
            error('*** Did Not Trim ***')
        end
    end
    error1=error0;
    k=k+1;
end

set(H_STATUS,'String','TRIMMING CYCLIC')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

t0=clock;
k=1;
error0=(((T/b)*rT1*(R-grip))*.04)+1;

while error0 > ((T/b)*rT1*(R-grip))*.04
    time=etime(clock,t0);
    if time > 15,
        set(H_STATUS,'String','STILL TRIMMING ...')
        set(H_STATUS1,'String',['Run Elapsed Time is '
num2str(fix(toc)) ' seconds'])
        pause(3)
        t0=clock;
    end

    Mpsi(:,k)=zeros(size(psi));
    tmcalc
    theta=[theta theta(:,k)];
    Mpsi=[Mpsi Mpsi(:,k)];

    %   *** calculation of initial dthetadM ***

    if k < 2,
        theta(:,k+1)=theta(:,k)+0.25/57.3;
        Mpsi(:,k+1)=zeros(size(psi));
        k=k+1;
```

```
    tmcalc
    k=k-1;
    dthetadM=(theta(:,k+1)-theta(:,k))./(Mpsi(:,k+1)-Mpsi(:,k));
end

%   *** calculation of M first harmonic parameters ***

M1c=2*sum(Mpsi(:,k).*cos(psi))/naz;
M1s=2*sum(Mpsi(:,k).*sin(psi))/naz;

%   *** removal of first harmonic terms from Mpsi ***

Mpsi(:,k+1)=Mpsi(:,k)-M1c.*cos(psi)-M1s.*sin(psi);
delM=Mpsi(:,k+1)-Mpsi(:,k);
error0=max(delM)-min(delM);
if k > 1,
    if error0 > error1,
        clc
        trim_warning
        set(GO,'Enable','off');
        set(RES,'Enable','off');
        set(RUPT,'Enable','off');
        set(BK,'Enable','on');
        error('*** END OF PROGRAM ***')
    end
end
error1=error0;

%   *** calculation of new theta ***

delM=0.5*(1-mu)*delM;
theta(:,k+1)=theta(:,k)+(dthetadM.*delM);

if error0 <= ((T/b)*rT1*(R-grip))*.04
    theta1c=2*sum(theta(:,k).*cos(psi))/naz;
    theta1s=2*sum(theta(:,k).*sin(psi))/naz;
else
    theta1c=2*sum(theta(:,k+1).*cos(psi))/naz;
    theta1s=2*sum(theta(:,k+1).*sin(psi))/naz;
end

theta(:,k+1)=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

%   *** calculation of new dthetadM ***

theta=[theta theta(:,k+1)];
Mpsi=[Mpsi Mpsi(:,k+1)];
theta(:,k+2)=theta(:,k)+0.25/57.3;
Mpsi(:,k+2)=zeros(size(Mpsi(:,k+1)));
k=k+2;
tmcalc
k=k-2;
dthetadM=(theta(:,k+2)-theta(:,k))./(Mpsi(:,k+2)-Mpsi(:,k));
k=k+1;
```

```
end

set(H_STATUS,'String','ADJUSTING COLLECTIVE')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

theta=theta(:,k);
k=1;
error0=(T*.01)+1;

while abs(error0) > T*.01
    Tpsi=zeros(size(psi));
    thrcalc
    error0=T-(mean(Tpsi)*b);
    if error0 < -T*.01,
        thetao=thetao-0.25*thetao*abs(1.25*error0/T)*(1-mu);
    elseif error0 > T*.01,
        thetao=thetao+0.25*thetao*abs(1.25*error0/T)*(1-mu);
    end
    theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);
    if k > 1,
        if abs(error0) > abs(error1),
            clc
            trim_warning
            error('*** END OF PROGRAM ***')
        end
    end
    error1=error0;
    k=k+1;
end

% *** calculating drag moments ***

DMpsi=zeros(size(psi));
dmcalc

%   *** calculating rotor H force ***

if Vinf < 16.9,
    Hrotor=0;
    dT=[dT ddT];
    dD=[dD ddD];
else
    dT=[dT ddT];
    dD=[dD ddD];
    for i=1:length(r)+1,
        H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;
        H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;
    end
    Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-
sin(betao)*sum(H1c)))+Drotor)/2;
end
```

```
%   *** calculating new rT ***

rT2=(((mean(Mpsi(:,length(Mpsi(1,:))-1))/mean(Tpsi))/R)+rT1)/2;

%   *** check rotor drag and rT, retrim rotor if required ***

while abs(Drotor-Hrotor) > 0.2*Hrotor | abs(rT1-rT2) > 0.015*rT1

    if abs(Drotor-Hrotor) > 0.2*Hrotor,
        set(H_STATUS,'String','ADJUSTING ROTOR DRAG')
        set(H_STATUS1,'String',['Run Elapsed Time is '
num2str(fix(toc)) ' seconds'])
        pause(3)
    end
    Drotor=Hrotor;
    if abs(rT1-rT2) > 0.015*rT1,
        set(H_STATUS,'String','ADJUSTING MEAN THRUST LOCATION')
        set(H_STATUS1,'String',['Run Elapsed Time is '
num2str(fix(toc)) ' seconds'])
        pause(2)
    end

    set(H_STATUS,'String','RETRIMMING ROTOR')
    set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
    pause(3)

    dT=dT(:,1:nbe);
    dD=dD(:,1:nbe);

    %   *** recalculating parameters ***

    alphaT=atan((Dftotal+Drotor)/(GW-Lftotal));
    mu=Vinf*cos(alphaT)/Vtip;

    if Vinf >= 16.9,    % Wheatley Eqn for Fwd flt
        T=(GW-Lftotal)/cos(alphaT);
        CT=T/(Adisk*rho*Vtip^2);
        lamdaT=0;
        lamda=1;
        while abs(lamdaT-lamda)>1e-4
            lamda=lamdaT;
            lamdaT=mu*sin(alphaT)+0.5*CT/sqrt(lamdaT^2+mu^2);
        end
        vi=lamdaT*Vtip-Vinf*sin(alphaT);
        vi=vi*ones(size(r));
    end

    B=1-(sqrt(2*CT)/b);
    Reff=B*R;
    Rbar=Reff-e;
    dr=(Reff-grip)/nbe;
    r=grip:dr:Reff-dr;,r=r+dr/2;
    RbarT=rT2*Rbar;
```

```
    betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade));

    %   *** trimming collective ***

    t0=clock;
    k=1;
    error0=(T*.02)+1;

    while abs(error0) > T*.02
        Tpsi=zeros(size(psi));
        thrcalc
        error0=T-(mean(Tpsi)*b);
        if error0 < -T*.02,
            thetao=thetao-0.35*thetao*abs(1.5*error0/T)*(1-mu);
        elseif error0 > T*.02,
            thetao=thetao+0.35*thetao*abs(1.5*error0/T)*(1-mu);
        end
        theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);
        if k > 1,
            if abs(error0) > abs(error1),
                clc
                trim_warning
                error('*** END OF PROGRAM ***')
            end
        end
        error1=error0;
        k=k+1;
    end

    %   *** trimming cyclic ***

    k=1;
    error0=(((T/b)*rT2*(R-grip))*.04)+1;

    while error0 > ((T/b)*rT2*(R-grip))*.04
        time=etime(clock,t0);
        if time > 15,
            set(H_STATUS,'String','STILL TRIMMING ...')
            set(H_STATUS1,'String',['Run Elapsed Time is '
num2str(fix(toc)) ' seconds'])
            pause(3)
            t0=clock;
        end

        Mpsi(:,k)=zeros(size(psi));
        tmcalc
        theta=[theta theta(:,k)];
        Mpsi=[Mpsi Mpsi(:,k)];

        %   *** calculation of initial dthetadM ***

        if k < 2,
            theta(:,k+1)=theta(:,k)+0.25/57.3;
```

```
            Mpsi(:,k+1)=zeros(size(psi));
            k=k+1;
            tmcalc
            k=k-1;
            dthetadM=(theta(:,k+1)-theta(:,k))./(Mpsi(:,k+1)-
Mpsi(:,k));
        end

        %   *** calculation of M first harmonic parameters ***

        M1c=2*sum(Mpsi(:,k).*cos(psi))/naz;
        M1s=2*sum(Mpsi(:,k).*sin(psi))/naz;

        %   *** removal of first harmonic terms from Mpsi ***

        Mpsi(:,k+1)=Mpsi(:,k)-M1c.*cos(psi)-M1s.*sin(psi);
        delM=Mpsi(:,k+1)-Mpsi(:,k);
        error0=max(delM)-min(delM);

        if k > 1,
            if error0 > error1,
                clc
                trim_warning
                error('*** END OF PROGRAM ***')
            end
        end

        error1=error0;

        %   *** calculation of new theta ***

        delM=0.5*(1-mu)*delM;
        theta(:,k+1)=theta(:,k)+(dthetadM.*delM);
        if error0 <= ((T/b)*rT2*(R-grip))*.04
            theta1c=2*sum(theta(:,k).*cos(psi))/naz;
            theta1s=2*sum(theta(:,k).*sin(psi))/naz;
        else
            theta1c=2*sum(theta(:,k+1).*cos(psi))/naz;
            theta1s=2*sum(theta(:,k+1).*sin(psi))/naz;
        end
        theta(:,k+1)=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

        %   *** calculation of new dthetadM ***


        theta=[theta theta(:,k+1)];
        Mpsi=[Mpsi Mpsi(:,k+1)];
        theta(:,k+2)=theta(:,k)+0.25/57.3;
        Mpsi(:,k+2)=zeros(size(Mpsi(:,k+1)));
        k=k+2;
        tmcalc
        k=k-2;
        dthetadM=(theta(:,k+2)-theta(:,k))./(Mpsi(:,k+2)-Mpsi(:,k));
        k=k+1;
```

```
    end

    %   *** retrimming collective ***

    theta=theta(:,k);
    k=1;
    error0=(T*.01)+1;

    while abs(error0) > T*.01
        Tpsi=zeros(size(psi));
        thrcalc
        error0=T-(mean(Tpsi)*b);
        if error0 < -T*.01,
            thetao=thetao-0.25*thetao*abs(1.25*error0/T)*(1-mu);
        elseif error0 > T*.01,
            thetao=thetao+0.25*thetao*abs(1.25*error0/T)*(1-mu);
        end
        theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);
        if k > 1,
            if abs(error0) > abs(error1),
                clc
                trim_warning
                error('*** END OF PROGRAM ***')
            end
        end
        error1=error0;
        k=k+1;
    end

    %   *** recalculating rotor H force ***

    if Vinf < 16.9,
        Hrotor=0;
        dT=[dT ddT];
        dD=[dD ddD];
    else
        dT=[dT ddT];
        dD=[dD ddD];
        for i=1:length(r)+1,
            H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;
            H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;
        end
        Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-
sin(betao)*sum(H1c)))+Drotor)/2;
    end

    %   *** recalculating rT ***

    rT1=rT2;
    rT2=(((mean(Mpsi(:,length(Mpsi(1,:))-1))/mean(Tpsi))/R)+rT1)/2;
end


% *** recalculating drag moments ***
```

```
dT=dT(:,1:nbe);
dD=dD(:,1:nbe);
DMpsi=zeros(size(psi));
dmcalc
dT=[dT ddT];
dD=[dD ddD];

set(H_STATUS,'String','ROTOR TRIMMED')
set(H_STATUS1,'String',['Run Elapsed Time is ' num2str(fix(toc)) '
seconds'])
pause(3)

save perftemp mchord DMpsi  % Eccles addition - perf.m was
                            % not recognizing mchord and DMpsi.
```

APPENDIX AH. THRCALC.M

This script M-file is a subroutine of Trim.m to calculate the rotor thrust.

```
%  Thrcalc.M
%  calculates the total thrust along a blade at
%  each azimuth (psi) location
%  JANRAD 98 VERSION 4.0

Up=zeros(size(psi*r));
Ut=zeros(size(Up));
dT=zeros(size(Up));
ddT=zeros(size(psi));

for i=1:length(psi),

Up(i,:)=vi.*cos(betao)+Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin
(betao)*cos(psi(i));
 Ut(i,:)=r.*omega+Vinf*cos(alphaT)*sin(psi(i));
 phi=atan2(Up(i,:),Ut(i,:));
 alpha=theta(i)+betat-phi;

% Eccles added the following line for use with Mach dependent afoil
files.
Mach =
(Vtip.*cos(alphaT).*r./R+Vinf.*sin(psi(i)))/(49.05*sqrt(temp+460));

 if afoil==1,
    [CL,CD]=oo12clcd(alpha, Mach);
 elseif afoil==2,
    [CL,CD]=hh02clcd(alpha);
 elseif afoil==3,
    [CL,CD]=vr12clcd(alpha);
 elseif afoil==4,
    [CL,CD]=sc1094r8clcd(alpha,Mach);
 elseif afoil==5,
    [CL,CD]=sc1095r8clcd(alpha,Mach);
 end

 dT(i,:)=0.5*rho.*cblade*dr.*(Up(i,:).^2+Ut(i,:).^2).*(CL.*cos(phi)-
CD.*sin(phi));
 Tpsi(i)=sum(dT(i,:));

%  *** calculations for tip loss area ***


Uptip=Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin(betao)*cos(psi(i
));
 Uttip=(R-(R-Reff)/2)*omega+Vinf*cos(alphaT)*sin(psi(i));
 phitip=atan2(Uptip,Uttip);
```

```
 ddT(i)=0.5*rho*cblade(nbe)*(0.5+0.5*cos(2*psi(i)))*(R-
Reff)*(Uptip^2+Uttip^2)*(-.009*sin(phitip));
 Tpsi(i)=Tpsi(i)+ddT(i);
end
```

APPENDIX AI. TMCALC.M

This script M-file is a subroutine of Trim.m to calculate the rotor thrust moment.

```
%  Tmcalc.m
%  calculates the total thrust moment along a blade
%  at each azimuth (psi) location
%  JANRAD 98 VERSION 4.0

Up=zeros(size(psi*r));
Ut=zeros(size(Up));
dM=zeros(size(Up));
ddM=zeros(size(psi));

for i=1:length(psi),

Up(i,:)=vi.*cos(betao)+Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin
(betao)*cos(psi(i));
 Ut(i,:)=r.*omega+Vinf*cos(alphaT)*sin(psi(i));
 phi=atan2(Up(i,:),Ut(i,:));
 alpha=theta(i,k)+betat-phi;

% Eccles added the following line for use with Mach dependent afoil
files.
 Mach =
(Vtip.*cos(alphaT).*r./R+Vinf.*sin(psi(i)))/(49.05*sqrt(temp+460));

 if afoil==1,
    [CL,CD]=oo12clcd(alpha, Mach);
 elseif afoil==2,
    [CL,CD]=hh02clcd(alpha);
 elseif afoil==3,
    [CL,CD]=vr12clcd(alpha);
 elseif afoil==4,
    [CL,CD]=sc1094r8clcd(alpha,Mach);
 elseif afoil==5,
    [CL,CD]=sc1095r8clcd(alpha,Mach);
 end

 dM(i,:)=0.5*rho.*cblade.*r*dr.*(Up(i,:).^2+Ut(i,:).^2).*(CL.*cos(phi)-
CD.*sin(phi));
 Mpsi(i,k)=sum(dM(i,:));

%  *** calculations for tip loss areas ***

Uptip=Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin(betao)*cos(psi(i
));
 Uttip=(R-(R-Reff)/2)*omega+Vinf*cos(alphaT)*sin(psi(i));
 phitip=atan2(Uptip,Uttip);
```

```
 ddM(i)=0.5*rho*cblade(nbe)*(0.5+0.5*cos(2*psi(i)))*(R-(R-Reff)/2)*(R-
Reff)*(Uptip^2+Uttip^2)*(-.009*sin(phitip));
 Mpsi(i,k)=Mpsi(i,k)+ddM(i);
end
```

APPENDIX AJ. DMCALC.M

This script M-file is a subroutine of Trim.m to calculate the rotor drag moment.

```
%  DMCALC.M
%  dmcalc calculates the total drag along a blade at
%  each azimuth (psi) location. Called from Trim.m
%  JANRAD 98 VERSION 4.0

Up=zeros(size(psi*r));
Ut=zeros(size(Up));
alpham=zeros(size(Up));
dD=zeros(size(Up));
ddD=zeros(size(psi));
ddDM=zeros(size(psi));

for i=1:length(psi),

Up(i,:)=vi.*cos(betao)+Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin
(betao)*cos(psi(i));
 Ut(i,:)=r.*omega+Vinf*cos(alphaT)*sin(psi(i));
 phi=atan2(Up(i,:),Ut(i,:));
 alpha=theta(i)+betat-phi;
 alpham(i,:)=alpha;

% Eccles added the following line for use with Mach dependent afoil
files.
 Mach =
(Vtip.*cos(alphaT).*r./R+Vinf.*sin(psi(i)))/(49.05*sqrt(temp+460));

 if afoil==1,
    [CL,CD]=oo12clcd(alpha, Mach);
 elseif afoil==2,
    [CL,CD]=hh02clcd(alpha);
 elseif afoil==3,
    [CL,CD]=vr12clcd(alpha);
 elseif afoil==4,
    [CL,CD]=sc1094r8clcd(alpha,Mach);
 elseif afoil==5,
    [CL,CD]=sc1095r8clcd(alpha,Mach);
 end


dD(i,:)=0.5*rho.*cblade*dr.*(Up(i,:).^2+Ut(i,:).^2).*(CL.*sin(phi)+CD.*
cos(phi));
 dDM=dD(i,:).*r;
 DMpsi(i)=sum(dDM);

%  *** calculations for tip loss area ***
```

```
Uptip=Vinf*sin(alphaT)*cos(betao)+Vinf*cos(alphaT)*sin(betao)*cos(psi(i
));
 Uttip=(R-(R-Reff)/2)*omega+Vinf*cos(alphaT)*sin(psi(i));
 phitip=atan2(Uptip,Uttip);

 ddD(i)=0.5*rho*cblade(nbe)*(0.5+0.5*cos(2*psi(i)))*(R-
Reff)*(Uptip^2+Uttip^2)*(.009*cos(phitip));
 ddDM(i)=ddD(i)*(R-(R-Reff)/2);
 DMpsi(i)=DMpsi(i)+ddDM(i);
end
```

APPENDIX AK. 0012CLCD.M

This MATLAB function calculates a c_l and c_d for a NACA 0012 airfoil given angle of attack and Mach number. It is called in Thrcalc.m, Tmcalc.m and Dmcalc.m script M-files.

```
%  oo12clcd calculates CL and CD for the NACA 0012
%  airfoil given angle of attack in radians and the
%  local Mach number:
%
% [CL,CD]=oo12clcd(alpha, Mach)
%
% Both 'alpha' and 'Mach' are intended to be vectors
% the elements of which correspond to the rotor blade
% radial stations of interest in a blade element analysis.
% All equations are based on Ray Prouty's treatment of
% the 0012 in his text.

function [CL,CD]=oo12clcd(alpha, Mach)
CL=zeros(size(alpha));
CD=zeros(size(alpha));
a=alpha*180/pi;
aL = 15 - 16.*Mach;
aD = 17 - 23.4.*Mach;
K1 = 0.0233 + 0.342.*(Mach.^7.15);
K2 = 2.05 - 0.95.*Mach;


% CL for Mach numbers < 0.725 and AOA inside +/- 20 deg:

chk=(Mach<0.725 & a>=0 & a<=aL);
 CL=CL+chk.*((0.1./sqrt(1-Mach.^2) - 0.01.*Mach).*a);

chk=(Mach<0.725 & a>aL & a<=20);
 CL=CL+chk.*((0.1./sqrt(1-Mach.^2) - 0.01.*Mach).*a - K1.*(a-aL).^K2);

chk=(Mach<0.725 & a>=-20 & a<-aL);
 CL=CL-chk.*((0.1./sqrt(1-Mach.^2) - 0.01.*Mach).*abs(a) - K1.*(abs(a)-
aL).^K2);

chk=(Mach<0.725 & a>=-aL & a<0);
 CL=CL-chk.*((0.1./sqrt(1-Mach.^2) - 0.01.*Mach).*abs(a));



% CL for Mach numbers > 0.725 and AOA inside +/- 20 deg:

chk=(Mach>=0.725 & a>=0 & a<=aL);
```

```
 CL=CL+chk.*((0.677 - 0.744.*Mach).*a);

chk=(Mach>=0.725 & a>aL & a<=20);
 CL=CL+chk.*((0.677 - 0.744.*Mach).*a - (0.0575-0.144.*(Mach-
0.725).^0.44).*(a-aL).^(K2));

chk=(Mach>=0.725 & a<0 & a>=-aL);
 CL=CL-chk.*((0.677 - 0.744.*Mach).*abs(a));

chk=(Mach>=0.725 & a<-aL & a>=-20);
 CL=CL-chk.*((0.677 - 0.744.*Mach).*abs(a) - (0.0575-0.144.*(Mach-
0.725).^0.44).*(abs(a)-aL).^(K2));


% CL for all Mach numbers and AOA outside +/- 20deg:

chk=(a>20 & a<=161);
 CL=CL+chk.*(1.15.*sin(2.*alpha));

chk=(a>161 & a<=173);
 CL=CL+chk.*(-0.7);

chk=(a>173 & a<=180);
 CL=CL+chk.*(0.1.*(a-180));

chk=(a>=-180 & a<=-173);
 CL=CL+chk.*(0.1.*(a+180));

chk=(a>-173 & a<=-161);
 CL=CL+chk.*(0.7);

chk=(a>-161 & a<-20);
 CL=CL+chk.*(1.15.*sin(2.*alpha));



% CD for Mach numbers < 0.725 and AOA inside +/- 20 deg:

chk=(Mach<0.725 & a>=0 & a<=aD);
 CD=CD+chk.*(0.0081 + (-350.*a + 396.*a.^2 - 63.3.*a.^3 +
3.66.*a.^4).*10.^(-6));

chk=(Mach<0.725 & a>aD & a<=20);
 CD=CD+chk.*((0.0081 + (-350.*a + 396.*a.^2 - 63.3.*a.^3 +
3.66.*a.^4).*10.^(-6)) + 0.00066.*(a-aD).^2.54);

chk=(Mach<0.725 & a<0 & a>=-aD);
 CD=CD+chk.*(0.0081 + (-350.*abs(a) + 396.*a.^2 - 63.3.*abs(a).^3 +
3.66.*a.^4).*10.^(-6));

chk=(Mach<0.725 & a<-aD & a>=-20);
 CD=CD+chk.*((0.0081 + (-350.*abs(a) + 396.*a.^2 - 63.3.*abs(a).^3 +
3.66.*a.^4).*10.^(-6)) + 0.00066.*(abs(a)-aD).^2.54);
```

```
% CD for Mach numbers > 0.725 and AOA inside +/- 20 deg:

chk=(Mach>=0.725 & a>=0 & a<=20);
 CD=CD+chk.*((0.0081 + (-350.*a + 396.*a.^2 - 63.3.*a.^3 +
3.66.*a.^4).*10.^(-6)) + 0.00035.*a.^2.54 + 21.*(Mach-0.725).^3.2);

chk=(Mach>=0.725 & a<0 & a>=-20);
 CD=CD+chk.*((0.0081 + (-350.*abs(a) + 396.*a.^2 - 63.3.*abs(a).^3 +
3.66.*a.^4).*10.^(-6)) + 0.00035.*abs(a).^2.54 + 21.*(Mach-
0.725).^3.2);

% CD for all Mach numbers and AOA outside +/- 20deg:

chk=(a>20 & a<=180);
 CD=CD+chk.*(1.03 - 1.02.*cos(2.*alpha));

chk=(a>=-180 & a<-20);
 CD=CD+chk.*(1.03 - 1.02.*cos(2.*alpha));
```

APPENDIX AL. HH02CLCD.M

This MATLAB function calculates a c_l and c_d for an HH-02 airfoil given angle of attack. It is called in Thrcalc.m, Tmcalc.m and Dmcalc.m script M-files.

```
function [CL,CD]=hh02clcd(alpha)

%  hh02clcd calculates CL and CD for an HH-02 airfoil
%  given angle of attack (alpha) in radians
%  [CL,CD]=hh02clcd(alpha)
%  JANRAD 98 VERSION 4.0

CL=zeros(size(alpha));
CD=zeros(size(alpha));
a=alpha*180/pi;

chk1=(a>=20 & a<=180);
 CL=CL+chk1.*(0.42541+0.026863*a+5.5988e-4*a.^2-2.1493e-5*a.^3+1.5932e-
7*a.^4-3.4659e-10*a.^5);
 CD=CD+chk1.*(-0.7179+0.061213*a-5.9861e-4*a.^2+7.3708e-6*a.^3-6.6605e-
8*a.^4+1.913e-10*a.^5);

chk1=(a>=-180 & a<=-50);
 CL=CL+chk1.*(-4.6183-0.1923*a-3.5554e-3*a.^2-3.3273e-5*a.^3-1.4528e-
7*a.^4-2.3003e-10*a.^5);
 CD=CD+chk1.*(2.7093e-2-2.1309e-2*a+2.0335e-4*a.^2+3.47e-7*a.^3-
3.0586e-8*a.^4-1.2584e-10*a.^5);

chk1=(a>-50 & a<-20);
 CL=CL+chk1.*(-2.5519-0.22847*a-9.5667e-3*a.^2-1.7051e-4*a.^3-1.0909e-
6*a.^4);
 CD=CD+chk1.*(2.7093e-2-2.1309e-2*a+2.0335e-4*a.^2+3.47e-7*a.^3-
3.0586e-8*a.^4-1.2584e-10*a.^5);

chk1=(a>=-20 & a<=-10);
 CL=CL+chk1.*(-0.2+0.089*a+0.0034*a.^2);
 CD=CD+chk1.*(2.7093e-2-2.1309e-2*a+2.0335e-4*a.^2+3.47e-7*a.^3-
3.0586e-8*a.^4-1.2584e-10*a.^5);

chk1=(a<20 & a>-10);
  CL=CL+chk1.*(5.8766e-2+1.3131e-1*a+2.4742e-3*a.^2-5.303e-4*a.^3-
1.5818e-5*a.^4+1.28e-6*a.^5);
  chk2=a<-4;
  chk2=chk2.*chk1;
   CD=CD+chk2.*(1.3786+0.916*a+0.21396*a.^2+2.0371e-2*a.^3+7.0076e-
4*a.^4);
  chk2=(a>=-4 & a<=7);
   chk2=chk2.*chk1;
```

```
   CD=CD+chk2.*(9.732e-3+3.2326e-4*a+1.4392e-4*a.^2-8.5073e-
5*a.^3+1.1826e-6*a.^4+1.5271e-6*a.^5);
  chk2=a>7;
  chk2=chk2.*chk1;
   CD=CD+chk2.*(1.842e-1-5.7532e-2*a+5.8043e-3*a.^2-1.2803e-4*a.^3);
```

APPENDIX AM. VR12CLCD.M

This MATLAB function calculates a c_l and c_d for an VR-12 airfoil given angle of attack. It is called in Thrcalc.m, Tmcalc.m and Dmcalc.m script M-files.

```
function [CL,CD]=vr12clcd(alpha)

%  vr12clcd calculates CL and CD for the VR-12 airfoil
%  given angle of attack (alpha) in radians
%  [CL,CD]=vr12clcd(alpha)
%  JANRAD 98 VERSION 4.0

CL=zeros(size(alpha));
CD=zeros(size(alpha));
a=alpha*180/pi;

chk=(a>=20 & a<=180);
 CL=CL+chk.*(1.1733-0.018879*a+1.5762e-3*a.^2-3.1925e-5*a.^3+2.0949e-
7*a.^4-4.3807e-10*a.^5);

chk=(a>=-180 & a<=-50);
 CL=CL+chk.*(-4.6183-0.1923*a-3.5554e-3*a.^2-3.3273e-5*a.^3-1.4528e-
7*a.^4-2.3003e-10*a.^5);

chk=(a>-50 & a<-30);
 CL=CL+chk.*(-0.22114+0.020857*a+2.8571e-4*a.^2);

chk=(a>=-30 & a<=-10);
 CL=CL+chk.*(-1.11-0.12383*a-0.01515*a.^2-6.8667e-4*a.^3-1e-5*a.^4);

chk=(a<20 & a>-10);
 CL=CL+chk.*(0.11976+0.12341*a+5.5841e-4*a.^2-2.0652e-4*a.^3);

chk=(a>=17 & a<=180);
 CD=CD+chk.*(-0.26376+0.017917*a+6.9927e-4*a.^2-9.1137e-6*a.^3+2.6277e-
8*a.^4);

chk=(a>=-180 & a<=-10);
 CD=CD+chk.*(-0.17486-0.034463*a-1.0233e-4*a.^2-2.8958e-6*a.^3-4.6577e-
8*a.^4-1.5557e-10*a.^5);

chk=(a>-10 & a<=0);
 CD=CD+chk.*(9.8678e-3+3.4934e-3*a+1.4844e-3*a.^2-1.3564e-4*a.^3-
1.0936e-5*a.^4);

chk=(a>0 & a<=15);
 CD=CD+chk.*(9.8e-3+7.0457e-4*a+5.6104e-5*a.^2-4.1151e-5*a.^3+3.8695e-
6*a.^4);
```

```
chk=(a>15 & a<17);
 CD=CD+chk.*(-1.33+1.325e-1*a-2.5e-3*a.^2);
```

APPENDIX AN. SC1094R8CLCD.M

This MATLAB function calculates a c_l and c_d for an Sikorsky SC1094R8 airfoil given angle of attack and Mach number. It is called in Thrcalc.m, Tmcalc.m and Dmcalc.m script M-files.

```
function [CL,CD]=Sc1094r8clcd(alpha,Mach)

%  sc1094r8 calculates CL and CD for a Sikorsky SC1094R8 airfoil
%  given angle of attack (alpha) in radians and Mach number (Mach)
%  [CL,CD]=sc1094r8clcd(alpha,Mach)

CL=zeros(size(alpha));
CD=zeros(size(alpha));


cl_tab=[0       0.     0.3     0.4     0.5     0.6     0.7     0.8      0.9
1.0;
        -180.  0.      0.      0.      0.      0.      0.      0.      0.
0.;
        -178.  .205    .205    .205    .205    .205    .205    .205    .205
.205;
        -176.  .410    .410    .41     .41     .41     .41     .41     .41
.41;
        -174.  .6      .6      .6      .6      .6      .6      .6      .6
.6;
        -172.  .77     .77     .77     .77     .77     .77     .77     .77
.77;
        -170.  .82     .82     .82     .82     .82     .82     .82     .82
.82;
        -168.  .82     .82     .82     .82     .82     .82     .82     .82
.82;
        -166.  .80     .8      .8      .8      .8      .8      .8      .8
.8;
        -164.  .76     .76     .76     .76     .76     .76     .76     .76
.76;
        -162.  .705    .705    .705    .705    .705    .705    .705    .705
.705;
        -160.  .65     .65     .65     .65     .65     .65     .65     .65
.65;
        -158.  .65     .65     .65     .65     .65     .65     .65     .65
.65;
        -90.   -.0627 -.0627 -.0627 -.0627 -.0627 -.0627 -.0627 -.0627
-.0627;
        -22.   -.98    -.98    -.775   -.827   -.84    -.926   -.875   -.838
-.822;
        -20.   -.975   -.975   -.65    -.765   -.773   -.92    -.856   -.81
-.79;
```

```
        -18.    -.689   -.689   -.527   -.704   -.707   -.914   -.838   -.782
-.758;
        -16.    -.403   -.403   -.403   -.642   -.64    -.908   -.819   -.754
-.726;
        -14.    -.278   -.278   -.28    -.58    -.573   -.839   -.8     -.716
-.694;
        -12.    -.242   -.242   -.343   -.57    -.522   -.708   -.79    -.698
-.662;
        -10.    -.72    -.72    -.48    -.56    -.488   -.576   -.81    -.67
-.630;
        -8.     -.63    -.63    -.52    -.53    -.453   -.49    -.75    -.666
-.622;
        -6.     -.49    -.49    -.47    -.49    -.418   -.45    -.69    -.663
-.615;
        -4.     -.29    -.29    -.27    -.295   -.315   -.36    -.47    -.487
-.428;
        -2.     -.07    -.07    -.047   -.085   -.11    -.105   -.25    -.310
-.24;
         0.      .15     .15     .176    .141    .145    .205    .07     -.150
-.05;
         2.      .37     .37     .399    .384    .425    .535    .35     .138
.2;
         4.      .59     .59     .622    .627    .72     .8      .56     .39
.449;
         6.      .81     .81     .845    .87     .89     .86     .705    .64
.7;
         8.     1.03    1.03    1.068   1.03     .939    .9      .805    .765
.806;
        10.     1.25    1.25    1.27    1.07     .976    .925    .841    .81
.85;
        12.     1.38    1.38    1.32    1.1     1.013    .93     .844    .83
.865;
        14.     1.5     1.5     1.078   1.117   1.05     .93     .848    .85
.88;
        16.     1.55    1.55    1.074   1.13    1.044    .92     .86     .87
.895;
        18.     1.072   1.072   1.07    1.07    1.038    .9      .88     .89
.91;
        20.     1.04    1.04    1.058   1.064   1.031    .9      .9      .91
.925;
        22.     1.032   1.032   1.046   1.058   1.025    .92     .92     .93
.94;
        90.     .0627   .0627   .0627   .0627   .0627   .0627   .0627   .0627
.0627;
       158.    -.66    -.66    -.66    -.66    -.66    -.66    -.66    -.66
-.66;
       160.    -.655   -.655   -.655   -.655   -.655   -.655   -.655   -.655
-.655;
       162.    -.685   -.685   -.685   -.685   -.685   -.685   -.685   -.685
-.685;
       164.    -.730   -.730   -.730   -.730   -.730   -.730   -.730   -.730
-.730;
       166.    -.770   -.77    -.77    -.77    -.77    -.77    -.77    -.77
-.77;
```

```
        168.    -.8     -.8     -.8     -.8     -.8     -.8     -.8     -.8
-.8;
        170.    -.805   -.805   -.805   -.805   -.805   -.805   -.805   -.805
-.805;
        172.    -.79    -.79    -.79    -.79    -.79    -.79    -.79    -.79
-.79;
        174.    -.61    -.61    -.61    -.61    -.61    -.61    -.61    -.61
-.61;
        176.    -.42    -.42    -.42    -.42    -.42    -.42    -.42    -.42
-.42;
        178.    -.21    -.21    -.21    -.21    -.21    -.21    -.21    -.21
-.21;
        180.    0.      0.      0.      0.      0.      0.      0.      0.
0. ];

cd_tab=[0     0.     0.3     0.4     0.5     0.6     0.7     0.8     0.9
1.0;
   -180.   .02     .02     .02     .02     .02     .02     .02     .02     .02;
   -178.   .03     .03     .03     .03     .03     .03     .03     .03     .03;
   -176.   .05     .05     .05     .05     .05     .05     .05     .05     .05;
   -174.   .08     .08     .08     .08     .08     .08     .08     .08     .08;
   -172.   .11     .11     .11     .11     .11     .11     .11     .11     .11;
   -170.   .14     .14     .14     .14     .14     .14     .14     .14     .14;
   -168.   .185    .185    .185    .185    .185    .185    .185    .185    .185;
   -166.   .235    .235    .235    .235    .235    .235    .235    .235    .235;
   -164.   .250    .250    .25     .25     .25     .25     .25     .25     .25;
   -162.   .265    .265    .265    .265    .265    .265    .265    .265    .265;
   -160.   .295    .295    .295    .295    .295    .295    .295    .295    .295;
   -158.   .36     .36     .36     .36     .36     .36     .36     .36     .36;
   -135.   1.1945  1.1945  1.1945  1.1945  1.1945  1.1945  1.1945  1.1945
1.1945;
   -90.    2.022   2.022   2.022   2.022   2.022   2.022   2.022   2.022
2.022;
   -60.    1.662   1.662   1.662   1.662   1.662   1.662   1.662   1.662
1.662;
   -45.    1.194   1.194   1.194   1.194   1.194   1.194   1.194   1.194
1.194;
   -30.    .6      .6      .6      .6      .6      .6      .6      .6      .6;
   -22.    .433    .433    .436    .43     .414    .458    .479    .5
.5154;
   -20.    .384    .384    .3875   .38     .36     .415    .441    .467
.4855;
   -18.    .334    .334    .339    .33     .322    .372    .403    .433    .457;
   -16.    .285    .285    .2905   .28     .284    .329    .3655   .4      .428;
   -14.    .2358   .2358   .242    .23     .246    .286    .3278   .367    .399;
   -12.    .1865   .1865   .1935   .193    .1973   .243    .29     .33     .37;
   -10.    .1373   .1373   .145    .15     .153    .162    .225    .262    .297;
   -8.     .088    .088    .091    .097    .102    .117    .17     .21     .248;
   -6.     .042    .042    .044    .051    .057    .076    .122    .163    .202;
   -4.     .013    .013    .013    .0215   .028    .04     .075    .115    .152;
   -2.     .008    .008    .008    .01     .01     .011    .028    .066    .117;
   0.      .0085   .0085   .008    .008    .008    .009    .025    .06     .1;
   2.      .009    .009    .008    .008    .008    .014    .07     .1      .136;
   4.      .01     .01     .009    .009    .015    .053    .108    .138    .17;
```

```
   6.     .0105  .0105  .0105  .013   .042   .092   .148   .182   .215;
   8.     .012   .012   .0125  .035   .083   .136   .182   .221   .255;
   10.    .014   .014   .0185  .071   .132   .1813  .225   .262   .298;
   12.    .016   .016   .05    .128   .189   .238   .285   .3225  .363;
   14.    .0225  .0225  .184   .2033  .238   .2816  .323   .357
.3925;
   16.    .085   .085   .2118  .257   .287   .325   .362   .391   .422;
   18.    .2054  .2054  .24    .31    .336   .368   .4     .425   .452;
   20.    .276   .276   .268   .363   .385   .412   .439   .459
.4815;
   22.    .347   .347   .295   .4167  .434   .456   .477   .493   .511;
   30.    .6     .6     .6     .6     .6     .6     .6     .6     .6;
   45.    1.194  1.194  1.194  1.194  1.194  1.194  1.194  1.194
1.194;
   60.    1.662  1.662  1.662  1.662  1.662  1.662  1.662  1.662
1.662;
   90.    2.022  2.022  2.022  2.022  2.022  2.022  2.022  2.022
2.022;
   135.   1.1945 1.1945 1.1945 1.1945 1.1945 1.1945 1.1945 1.1945
1.1945;
   158.   .36    .36    .36    .36    .36    .36    .36    .36    .36;
   160.   .295   .295   .295   .295   .295   .295   .295   .295   .295;
   162.   .265   .265   .265   .265   .265   .265   .265   .265   .265;
   164.   .250   .25    .25    .25    .25    .25    .25    .25    .25;
   166.   .235   .235   .235   .235   .235   .235   .235   .235   .235;
   168.   .185   .185   .185   .185   .185   .185   .185   .185   .185;
   170.   .140   .14    .14    .14    .14    .14    .14    .14    .14;
   172.   .110   .11    .11    .11    .11    .11    .11    .11    .11;
   174.   .08    .08    .08    .08    .08    .08    .08    .08    .08;
   176.   .05    .05    .05    .05    .05    .05    .05    .05    .05;
   178.   .03    .03    .03    .03    .03    .03    .03    .03    .03;
   180.   .02    .02    .02    .02    .02    .02    .02    .02    .02];

for j = 1:size(alpha)
   if alpha(:,j) < -pi
      alpha(:,j) = alpha(:,j) + 2*pi;
   elseif alpha(:,j) > pi
      alpha(:,j) = alpha(:,j) - 2*pi;
   end
end

Mach = abs(Mach);
a=alpha.*180/pi;

CL = diag(table2(cl_tab,a,Mach))';
CD = diag(table2(cd_tab,a,Mach))';
```

APPENDIX AO. SC1095R8CLCD.M

This MATLAB function calculates a c_l and c_d for an Sikorsky SC1095R8 airfoil given angle of attack and Mach number. It is called in Thrcalc.m, Tmcalc.m and Dmcalc.m script M-files.

```
function [CL,CD]=Sc1095r8clcd(ALPHA,MACH)

%  sc1095r8 calculates CL and CD for a Sikorsky SC1095R8 airfoil
%  given angle of attack (ALPHA) in radians and Mach number (mach)
%  [CL,CD]=sc1095r8clcd(ALPHA,MACH)

CL=zeros(size(ALPHA));
CD=zeros(size(ALPHA));

cl_tab=[   0    0.     .3     .4     .5     .6     .7     .8     .9
1.;
-180.  0.     0.     0.     0.     0.     0.     0.     0.     0.;
-178.  .205   .205   .205   .205   .205   .205   .205   .205   .205;
-176.  .41    .41    .41    .41    .41    .41    .41    .41    .41;
-174.  .6     .6     .6     .6     .6     .6     .6     .6     .6;
-172.  .77    .77    .77    .77    .77    .77    .77    .77    .77;
-170.  .82    .82    .82    .82    .82    .82    .82    .82    .82;
-168.  .82    .82    .82    .82    .82    .82    .82    .82    .82;
-166.  .8     .8     .8     .8     .8     .8     .8     .8     .8;
-164.  .76    .76    .76    .76    .76    .76    .76    .76    .76;
-162.  .705   .705   .705   .705   .705   .705   .705   .705   .705;
-160.  .65    .65    .65    .65    .65    .65    .65    .65    .65;
-158.  .65    .65    .65    .65    .65    .65    .65    .65    .65;
-90.   -.0627 -.0627 -.0627 -.0627 -.0627 -.0627 -.0627 -.0627 -.0627;
-22.   -.98   -.98   -.98   -.914  -.934  -.926  -.875  -.838  -.822;
-20.   -.975  -.975  -.96   -.910  -.93   -.920  -.856  -.81   -.79;
-18.   -.969  -.969  -.962  -.906  -.926  -.914  -.838  -.782  -.758;
-16.   -.963  -.963  -.966  -.902  -.922  -.908  -.819  -.754  -.726;
-14.   -1.07  -1.07  -.824  -.803  -.805  -.88   -.8    -.726  -.694;
-12.   -.718  -.718  -.532  -.528  -.66   -.83   -.79   -.698  -.662;
-10.   -.366  -.366  -.24   -.4    -.61   -.78   -.81   -.67   -.630;
-8.    -.245  -.245  -.3    -.33   -.55   -.74   -.75   -.666  -.622;
-6.    -.39   -.39   -.44   -.32   -.52   -.68   -.69   -.663  -.615;
-4.    -.4    -.4    -.42   -.44   -.47   -.58   -.47   -.486  -.428;
-2.    -.185  -.185  -.185  -.196  -.199  -.255  -.25   -.31   -.24;
0.     .029   .029   .048   .048   .072   .07    .07    -.15   -.05;
2.     .244   .244   .282   .292   .343   .395   .35    .138   .2;
4.     .459   .459   .515   .536   .614   .72    .56    .39    .449;
6.     .673   .673   .749   .78    .84    .83    .705   .64    .7;
8.     .888   .888   .983   .96    .91    .882   .805   .765   .806;
10.    1.103  1.103  1.17   1.01   .946   .92    .842   .81    .85;
12.    1.25   1.25   1.13   .96    1.     .924   .845   .829   .865;
```

```
14.     1.1     1.1     1.03    1.07    1.053   .928    .848    .848    .88;
16.     .98     .98     .96     1.06    1.075   .92     .860    .867    .895;
18.     .982    .982    .966    1.07    1.064   .9      .880    .886    .91;
20.     .984    .984    .972    1.065   1.053   .9      .900    .905    .925;
22.     .987    .987    .979    1.06    1.042   .92     .920    .924    .94;
90.     .0627   .0627   .0627   .0627   .0627   .0627   .0627   .0627   .0627;
158.    -.66    -.66    -.66    -.66    -.66    -.66    -.66    -.66    -.66;
160.    -.655   -.655   -.655   -.655   -.655   -.655   -.655   -.655   -.655;
162.    -.685   -.685   -.685   -.685   -.685   -.685   -.685   -.685   -.685;
164.    -.73    -.73    -.73    -.73    -.73    -.73    -.73    -.73    -.73;
166.    -.77    -.77    -.77    -.77    -.77    -.77    -.77    -.77    -.77;
168.    -.8     -.8     -.8     -.8     -.8     -.8     -.8     -.8     -.8;
170.    -.805   -.805   -.805   -.805   -.805   -.805   -.805   -.805   -.805;
172.    -.79    -.79    -.79    -.79    -.79    -.79    -.79    -.79    -.79;
174.    -.61    -.61    -.61    -.61    -.61    -.61    -.61    -.61    -.61;
176.    -.42    -.42    -.42    -.42    -.42    -.42    -.42    -.42    -.42;
178.    -.21    -.21    -.21    -.21    -.21    -.21    -.21    -.21    -.21;
180.    0.      0.      0.      0.      0.      0.      0.      0.      0.];

cd_tab=[ 0    0.      .3      .4      .5      .6      .7      .8      .9
1.;
-180.   .02     .02     .02     .02     .02     .02     .02     .02     .02;
-178.   .03     .03     .03     .03     .03     .03     .03     .03     .03;
-176.   .05     .05     .05     .05     .05     .05     .05     .05     .05;
-174.   .08     .08     .08     .08     .08     .08     .08     .08     .08;
-172.   .11     .11     .11     .11     .11     .11     .11     .11     .11;
-170.   .14     .14     .14     .14     .14     .14     .14     .14     .14;
-168.   .185    .185    .185    .185    .185    .185    .185    .185    .185;
-166.   .235    .235    .235    .235    .235    .235    .235    .235    .235;
-164.   .25     .25     .25     .25     .25     .25     .25     .25     .25;
-162.   .265    .265    .265    .265    .265    .265    .265    .265    .265;
-160.   .295    .295    .295    .295    .295    .295    .295    .295    .295;
-158.   .36     .36     .36     .36     .36     .36     .36     .36     .36;
-135.   1.1945  1.1945  1.1945  1.1945  1.1945  1.1945  1.1945  1.1945  1.1945;
-90.    2.022   2.022   2.022   2.022   2.022   2.022   2.022   2.022   2.022;
-60.    1.662   1.662   1.662   1.662   1.662   1.662   1.662   1.662   1.662;
-45.    1.194   1.194   1.194   1.194   1.194   1.194   1.194   1.194   1.194;
-30.    .6      .6      .6      .6      .6      .6      .6      .6      .6;
-22.    .3438   .3438   .3885   .4065   .414    .458    .479    .497    .514;
-20.    .2723   .2723   .3281   .3506   .36     .415    .441    .463    .486;
-18.    .2007   .2007   .2678   .2948   .3267   .372    .403    .43     .457;
-16.    .1292   .1292   .2073   .2388   .2887   .329    .3655   .397    .428;
-14.    .0576   .0576   .147    .183    .246    .286    .3278   .363    .399;
-12.    .0174   .0174   .0225   .12     .191    .243    .29     .33     .37;
-10.    .008    .008    .0132   .068    .127    .177    .225    .262    .297;
-8.     .0082   .0082   .0095   .0206   .07     .113    .16     .203    .248;
-6.     .0079   .0079   .0085   .0097   .026    .06     .1      .149    .202;
-4.     .0075   .0075   .008    .008    .0125   .03     .065    .115    .152;
-2.     .0075   .0075   .008    .0075   .0085   .012    .028    .066    .117;
0.      .0075   .0075   .008    .0075   .008    .008    .017    .05     .09;
2.      .008    .008    .0082   .0075   .0075   .0105   .04     .08     .1175;
4.      .0085   .0085   .0085   .008    .011    .036    .09     .12     .1525;
6.      .009    .009    .0105   .011    .029    .081    .128    .167    .203;
8.      .011    .011    .014    .026    .0743   .126    .17     .21     .249;
```

```
10.    .017   .017   .021   .08    .1247  .162   .225   .262   .298;
12.    .026   .026   .0935  .153   .18    .238   .285   .3225  .363;
14.    .145   .145   .1635  .2121  .246   .284   .326   .357   .393;
16.    .2147  .2147  .2259  .2643  .2887  .329   .3655  .391   .423;
18.    .274   .274   .2836  .3166  .3267  .327   .403   .43    .457;
20.    .3333  .3333  .3414  .3688  .36    .415   .441   .463   .486;
22.    .2927  .2927  .3991  .421   .414   .458   .479   .497   .514;
30.    .6     .6     .6     .6     .6     .6     .6     .6     .6;
45.    1.194  1.194  1.194  1.194  1.194  1.194  1.194  1.194  1.194;
60.    1.662  1.662  1.662  1.662  1.662  1.662  1.662  1.662  1.662;
90.    2.022  2.022  2.022  2.022  2.022  2.022  2.022  2.022  2.022;
135.   1.1945 1.1945 1.1945 1.1945 1.1945 1.1945 1.1945 1.1945 1.1945;
158.   .36    .36    .36    .36    .36    .36    .36    .36    .36;
160.   .295   .295   .295   .295   .295   .295   .295   .295   .295;
162.   .265   .265   .265   .265   .265   .265   .265   .265   .265;
164.   .25    .25    .25    .25    .25    .25    .25    .25    .25;
166.   .235   .235   .235   .235   .235   .235   .235   .235   .235;
168.   .185   .185   .185   .185   .185   .185   .185   .185   .185;
170.   .14    .14    .14    .14    .14    .14    .14    .14    .14;
172.   .11    .11    .11    .11    .11    .11    .11    .11    .11;
174.   .08    .08    .08    .08    .08    .08    .08    .08    .08;
176.   .05    .05    .05    .05    .05    .05    .05    .05    .05;
178.   .03    .03    .03    .03    .03    .03    .03    .03    .03;
180.   .02    .02    .02    .02    .02    .02    .02    .02    .02];


for j = 1:size(alpha)
   if alpha(:,j) < -pi
      alpha(:,j) = alpha(:,j) + 2*pi;
   elseif alpha(:,j) > pi
      alpha(:,j) = alpha(:,j) - 2*pi;
   end
end

Mach = abs(Mach);
a=alpha.*180/pi;

CL = diag(table2(cl_tab,a,Mach))';
CD = diag(table2(cd_tab,a,Mach))';
```

APPENDIX AP. VARIABLE_LIST.M

This script M-file contains a list of variables used in JANRAD 98 version 4.0. The list is broken up into four parts. The first part lists variables used primarily for computation and analysis. Then global variables, structure variables and GUI graphic handles are listed.

```
%  This File lists the Variables used in JANRAD 98  Version 4.0

%  Computational Vars.

%   a           lift curve slope of rotor system airfoil
%   Adisk       area of rotor disk
%   Afh         fuselage equivalent flat plate drag area
%   Afv         vertical projected area (fuselage area under disk)
%   afoil       rotor system airfoil type (HH02/VR12)
%   alpha       angle of attack, rotor blade radial segment
%   alphaT      rotor tip path plane angle
%   b           number of rotor blades
%   B           tip loss parameter
%   betao       rotor coning angle
%   betat       geometric angle, rotor blade radial segment
%   bhoriz      span, horizontal tail
%   bvert       span, vertical tail
%   bwing       span, wing
%   cblade      chord, rotor blade
%   CD          drag coefficient, rotor blade radial segment
%   CDohoriz    profile drag coefficient, horizontal tail
%   CDovert     profile drag coefficient, vertical tail
%   CDowing     profile drag coefficient, wing
%   CDhoriz     drag coefficient, horizontal tail
%   CDvert      drag coefficient, vertical tail
%   CDwing      drag coefficient, wing
%   CH          rotor H-force coefficient
%   CH_sig      CH/solidity
%   CL          lift coefficient, rotor blade radial segment
%   CLhoriz     lift coefficient, horizontal tail
%   CLvert      lift coefficient, vertical tail
%   CLwing      lift coefficient, wing
%   CQ          rotor torque coefficient
%   CQ_sig      CQ/solidity
%   CT          rotor thrust coefficient
%   CT_sig      CT/solidity
%   dD          differential drag, rotor blade radial segment
%   ddD         differential drag, rotor blade tip
%   ddDM        differential drag moment, rotor blade tip
%   ddM         differential thrust moment, rotor blade tip
```

```
%   ddT         differential thrust, rotor blade tip
%   delM        change in total thrust moment
%   Dftotal     resultant of fuselage drag and aux thrust
%   Dfuse       total drag generated by non-rotor bodies
%   DL          disk loading
%   dM          differential thrust moment, rotor blade radial seg
%   DMpsi       total blade drag moment at specific azimuth angle
%   dr          rotor blade radial segment width
%   Drotor      rotor system drag
%   dT          differential thrust, rotor blade radial segment
%   Dhoriz      drag, horizontal tail
%   dthetadM    change in cyclic pitch with change in thrust moment
%   Dvert       drag, vertical tail
%   Dwing       drag, wing
%   e           effective hinge offset
%   ewing       wing efficiency factor
%   filename    name of input file
%   FM          figure of merit
%   grip        length of inner non-aerodynamic portion of blade
%   GW          aircraft gross weight
%   Hrotor      rotor H-force
%   lamdaT      forward flight induced velocity parameter
%   Lftotal     total lift generated by non-rotor bodies
%   Lhoriz      lift, horizontal tail
%   Lvert       lift, vertical tail
%   Lwing       lift, wing
%   M1c         first harmonic (cosine) thrust moment coefficient
%   M1s         first harmonic (sine) thrust moment coefficient
%   Machtip     Mach number at rotor blade tip
%   mblade      mass of rotor blade
%   Mpsi        total blade thrust moment at specific azimuth angle
%   mu          advance ratio
%   naz         number of azimuth sectors
%   nbe         number of blade elements
%   omega       rotor rotational velocity
%   PA          pressure altitude
%   phi         inflow angle, rotor blade radial segment
%   phitip      inflow angle, rotor blade tip
%   Protor      power required by rotor
%   psi         azimuth angle
%   q           dynamic pressure
%   Qrotor      rotor torque
%   r           radius, rotor blade radial segment
%   R           rotor blade radius
%   Rbar        Reff-e
%   RbarT       rT*Rbar
%   Reff        effective rotor blade radius (tip loss)
%   rho         ambient air density
%   rT          location of resultant thrust vector
%   solidity    solidity
%   Shoriz      area, horizontal tail
%   Svert       area, vertical tail
%   Swing       area, wing
%   T           rotor thrust
```

```
%   Taux        auxiliary thrust
%   temp        ambient air temperature
%   theta       cyclic pitch
%   thetalc     first harmonic (cosine) of cyclic pitch
%   thetals     first harmonic (sine) of cyclic pitch
%   thetao      collective pitch at .7 r/R
%   Tpsi        total blade thrust at specific azimuth angle
%   tr          rotor blade taper ratio
%   twist       geometric rotor blade twist
%   Up          vertical component of velocity
%   Uptip       vertical component of velocity at tip
%   Ut          horizontal component of velocity
%   Uttip       horizontal component of velocity at tip
%   vi          induced velocity
%   Vinf        forward airspeed
%   Vtip        tip speed
%   wblade      weight of rotor blade

%  Global Vars.

%  COUNT           Counter to determine where Performance Input was
called
%  NAME            Input .mat file name
%  PICK            Iteration Method Choice (1-7)
%  REGIME          Include HIGE Calculations Choice (1=yes, 0=no)
%  MINUM           Iteration Start Value
%  MAXUM           Iteration End Value
%  INTER           Iteration Interval

%  Structure Vars.

%  S_MATR_VEC      Matrix/Vector structure
%  S_PERF_INPUT    Perf.m input structure
%  S_USER_INPUT    user input structure

%  Graphics Handle Vars.

%  H_AL            Altitude Iteration Radio Button
%  H_ANAL          Analysis Figure Window
%  H_AS            Airspeed Iteration Radio Button
%  H_ASPECT        Aspect Ratio Static Text Box
%  H_ASPECT_EDIT   Aspect Ratio Edit Text Box
%  H_BK            Iteration Method << Back Push Button
%  H_BT            Blade Twist Iteration Radio Button
%  H_BTR           Blade Taper Ratio Iteration Radio Button
%  H_check1        Save Input Data Check box
%  H_check2        Save Output Data Check box
%  H_check3        Save Matrix & Vector Data Check box
%  H_CID           Change Input Data Radio Button
%  H_CIM           Change Iteration Method Radio Button
%  H_CNF           Create New Radio Button
%  H_datain        Save Input Data Edit Box
%  H_dataout       Save Output Data Edit Box
%  H_DISK          Horiz. Tail Under Main Rotor Disk Check Box
```

```
%  H_EJANRAD       Exit JANRAD Radio Button
%  H_EREF          Edit/Run Existing File Radio Button
%  H_GO            Analyze Push Button
%  H_GW            Gross Weight Iteration Radio Button
%  H_HIGE          Iteration Parameters HIGE Check box
%  H_inputfile     Input File Static Text box
%  H_IP            Iteration Parameters figure window
%  H_IT_BOX        Iteration Parameters Static Text Box
%  H_IT_METH       Iteration Method figure window
%  H_JAN           JANRAD 98 Figure window
%  H_LB            Input File List Box
%  H_MEN           JANRAD 98 Options Menu handle
%  H_NI            No Iteration Radio Button
%  h_opt           Performance Output JANRAD Options Menu
%  H_OPTIONS       Options Figure Window
%  H_outputfile    Output File Static Text box
%  H_P             Performace Radio Button
%  H_PERF_IN       Performance Input Figure Window
%  H_PERF_OUT      Performance Output Figure Window
%  H_PRDA          Perform Rotor Dynamics Radio Button
%  H_printin       Print Input File Check Box
%  H_printout      Print Output File Check Box
%  H_printvec      Print Matrix & Vector File Check Box
%  H_PSCA          Perform Stability and Control Radio Button
%  H_RD            Rotor Dynamics Radio Button
%  H_RES           Resume Push Button
%  H_RTB           Return to Beginning Radio Button
%  H_RUPT          Interrupt Push Button
%  H_SAC           Stability and Control Radio Button
%  H_SOT           Start of Taper Iteration Radio Button
%  H_STATUS        Top Analysis Status Static Text Box
%  H_STATUS1       Middle Analysis Status Static Text Box (Elapsed Time)
%  H_STATUS2       Bottom Analysis Status Static Text Box (not used yet)
%  H_vecdata       Save Matrix & Vector Data Edit Box
%  H_vecfile       Vector File Static Text box
%  H_WORK          Working Directory Edit Box
%  H_WSA           Wing Span Area Iteration Radio Button
```

LIST OF REFERENCES

Dean, L., Building *a Graphical User Interface with MATLAB® 5,* The MathWorks Inc., 1997.

Dean, L., The Design of Callbacks, MATLAB® Digest, Volume 5, No.3, December 1997.

Eccles, D. M., A *Validation of the Joint Army/Navy Rotorcraft Analysis and Design Software by Comparison with H-34 and UUH-60A Flight Test,* Naval Postgraduate School, Thesis for MSAE Degree, Monterey, Ca, 1995.

Hanselman, D. C. / Littlefeild B., *Mastering MATLAB®*, Prentice Hall, 1996.

Marchand, P., *Graphics and GUI's with MATLAB®*, CRC Press, 1996.

MATLAB® 5.1 Software Documentation, *Building GUI's with MATLAB®*, The MathWorks Inc., 1997.

MATLAB® 5.1 Software Documentation, *Using MATLAB®*, ch. 13, The MathWorks Inc., 1997.

Nicholson, R. K. Jr., Computer *Code for Interactive Rotorcraft Preliminary Design Using a Harmonic Balance Method for Rotor Trim,* Naval Postgraduate School, Thesis for MSAE Degree, Monterey, Ca, 1993.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, Virginia 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, California 93943-5101	2
3.	Professor Gerald Lindsey, Code AA/Li Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor E. Roberts Wood, Code AA/Wd Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	2
5.	LCDR. Robert L. King, USN, Code AA/Ki Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93943-5000	1
6.	LCDR. Chris F. Lapacik, USN 101 Moreell Circle Monterey, California 93940-4828	2
7.	LCDR. William Hucke, USCG 22 Mervine Street Monterey, California 93940-4828	1
8.	Dr. Greg Severn Department of Physics University of San Diego 5998 Alcala Park San Diego, California 92110	1
9.	Mr. Raymond Prouty 4224 Deerpark Court Westlake Village, California 91361	1

10. Loren Dean 1
The MathWorks, Inc.
24 Prime Park Way
Natick, Massachusetts 01760-150010

11. Dr. Michael P. Scully 1
P.O. Box 1120
Mountain View, California 94042